

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-359107

(43)Date of publication of application : 26.12.2001

(51)Int.Cl. H04N 7/32
H03M 7/36
H04N 5/92
H04N 11/04

(21)Application number : 2001-114698

(71)Applicant : SONY CORP

(22)Date of filing : 13.04.2001

(72)Inventor : UEDA MAMORU
KANESAKA KOKI
OHARA TAKUMI
YAMAMOTO TAKESHI
MIZUNO KAZUHIRO
MORITA TOSHIKAZU

(30)Priority

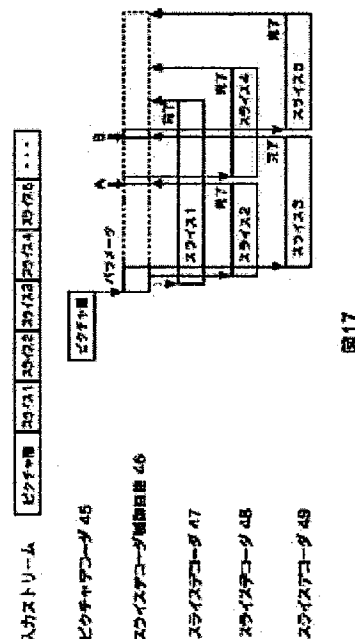
Priority number : 2000112951 Priority date : 14.04.2000 Priority country : JP

(54) DECODER AND DECODING METHOD, RECORDING MEDIUM, AND PROGRAM

(57)Abstract:

PROBLEM TO BE SOLVED: To control operations of a plurality of slice decoders.

SOLUTION: A slice decoder control circuit 46 receiving an input of a parameter of a picture layer sequentially supplies a parameter of the picture layer and a write pointer of a slice 1 to a slice decoder 47, supplies the parameter of the picture layer and a write pointer of a slice 2 to a slice decoder 48, and supplies the parameter of the picture layer and a write pointer of a slice 3 to a slice decoder 49 respectively so as to allow the decoders to decode the slices of an input stream. The slice decoder control circuit 46 supplies a write pointer of a slice 4 to the slice decoder 48 in timing A to allow the slice decoder 48 to decode the slice 4, supplies a write pointer of a slice 5 to the slice decoder 49 in timing B to allow the slice decoder 49 to decode the slice 5 and repeats the similar processing until decoding of the final



slice is finished.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2001-359107
(P2001-359107A)

(43) 公開日 平成13年12月26日 (2001. 12. 26)

(51) Int.Cl. ⁷	識別記号	F I	テマコード* (参考)
H 0 4 N 7/32		H 0 3 M 7/36	5 C 0 5 3
H 0 3 M 7/36		H 0 4 N 11/04	A 5 C 0 5 7
H 0 4 N 5/92		7/137	Z 5 C 0 5 9
11/04		5/92	H 5 J 0 6 4

審査請求 未請求 請求項の数28 O L (全 33 頁)

(21) 出願番号 特願2001-114698(P2001-114698)
(22) 出願日 平成13年4月13日 (2001. 4. 13)
(31) 優先権主張番号 特願2000-112951(P2000-112951)
(32) 優先日 平成12年4月14日 (2000. 4. 14)
(33) 優先権主張国 日本 (J P)

(71) 出願人 000002185
ソニー株式会社
東京都品川区北品川 6 丁目 7 番35号
(72) 発明者 上田 衛
東京都品川区北品川 6 丁目 7 番35号 ソニ
ー株式会社内
(72) 発明者 金坂 幸喜
神奈川県川崎市麻生区南黒川10番1号 株
式会社エルエスアイシステムズ内
(74) 代理人 100082131
弁理士 稲本 義雄

最終頁に続く

(54) 【発明の名称】 復号装置および復号方法、記録媒体、並びにプログラム。

(57) 【要約】

【課題】 複数のスライスデコーダの動作を制御する。
【解決手段】 ピクチャ層のパラメータの入力を受けたスライスデコーダ制御回路46は、スライスデコーダ47にピクチャ層のパラメータとスライス1の書き込みポイントを、スライスデコーダ48にピクチャ層のパラメータとスライス2の書き込みポイントを、スライスデコーダ49にピクチャ層のパラメータとスライス3の書き込みポイントを、それぞれ順番に供給してデコードさせる。スライスデコーダ制御回路46は、スライスデコーダ47乃至49から入力されるデコード処理の完了を示す信号の入力を基に、タイミングAで、スライスデコーダ48にスライス4の書き込みポイントを供給してデコードさせ、タイミングBで、スライスデコーダ49にスライス5の書き込みポイントを供給してデコードさせ、以下、最後のスライスのデコードが終了されるまで同様の処理が繰り返される。

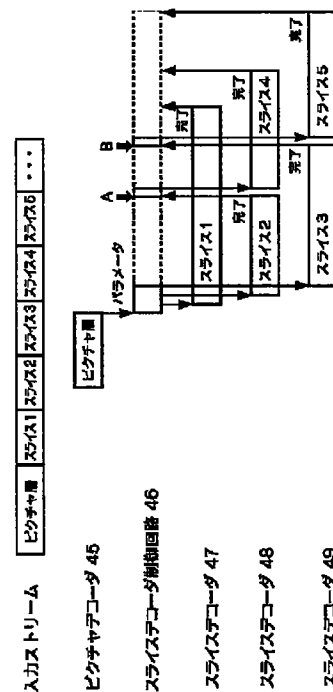


図17

【特許請求の範囲】

【請求項1】 符号化ストリームを復号する復号装置において、

前記符号化ストリームを復号する複数の復号手段と、
複数の前記復号手段を並行して動作させるように制御する復号制御手段とを備えることを特徴とする復号装置。

【請求項2】 複数の前記復号手段は、復号処理の終了を示す信号を前記復号制御手段に出力し、

前記復号制御手段は、復号処理の終了を示す前記信号を出力した前記復号手段に、新たな前記符号化ストリームを復号させるように制御することを特徴とする請求項1に記載の復号装置。

【請求項3】 前記符号化ストリームをバッファリングする第1のバッファ手段と、

前記符号化ストリームから、前記符号化ストリームに含まれる所定の情報の単位の始まりを表わすスタートコードを読み出すとともに、前記第1のバッファ手段に、前記スタートコードが保持されている位置に関する位置情報を読み出す読み出し手段と、

前記読み出し手段により読み出された、前記スタートコードおよび前記位置情報をバッファリングする第2のバッファ手段と、

前記第1のバッファ手段による前記符号化ストリームのバッファリング、および前記第2のバッファ手段による前記スタートコードおよび前記位置情報のバッファリングを制御するバッファリング制御手段とを更に備えることを特徴とする請求項1に記載の復号装置。

【請求項4】 前記符号化ストリームは、ISO/IEC13818-2およびITU-T勧告H.262に規定されたMPEG2の符号化ストリームであることを特徴とする請求項1に記載の復号装置。

【請求項5】 複数の前記復号手段により復号され、出力された複数の画像データのうちの所定のものを選択する選択手段と、

前記選択手段により選択された前記画像データの入力を受け、必要に応じて動き補償を施す動き補償手段とを更に備えることを特徴とする請求項1に記載の復号装置。

【請求項6】 前記復号手段は、復号処理が終了したことを示す終了信号を前記選択手段に出力し、

前記選択手段は、
複数の前記復号手段のそれぞれの処理状態に対応する値を記憶する記憶手段を有し、

前記記憶手段の値が全て第1の値になった場合、復号処理が終了したことを示す前記終了信号を出力している前記復号手段に対応する前記記憶手段に記憶されている値を、前記第1の値から第2の値に変更し、

対応する前記記憶手段に記憶されている値が前記第2の値である前記第1の復号手段により復号された前記画像データのうち、いずれかの前記画像データを選択し、

選択された前記画像データを復号した前記復号手段に対

応する前記記憶手段に記憶されている値を前記第1の値に変更することを特徴とする請求項5に記載の復号装置。

【請求項7】 前記復号手段により復号された前記画像データ、前記選択手段により選択された前記画像データ、または前記動き補償手段により動き補償が施された前記画像データを保持する保持手段と、

前記選択手段により選択された前記画像データ、または前記動き補償手段により動き補償が施された前記画像データの前記保持手段による保持を制御する保持制御手段とを更に備えることを特徴とする請求項5に記載の復号装置。

【請求項8】 前記保持手段は、前記画像データの輝度成分と色差成分をそれぞれ分けて保持することを特徴とする請求項7に記載の復号装置。

【請求項9】 前記復号手段に供給される前記符号化ストリームのフレームの順番を変更する変更手段を更に備え、

前記保持手段は、画像シーケンス内のイントラ符号化フレームおよび前方向予測符号化フレームを合計したフレーム数より少なくとも2つ多い数のフレームを保持することができ、

前記変更手段は、前記符号化ストリームを逆転再生させるための所定の順番になるように、前記符号化ストリームのフレームの順番を変更することを特徴とする請求項7に記載の復号装置。

【請求項10】 前記保持手段により保持されている前記画像データを読み出して出力する出力手段を更に備え、

前記所定の順番とは、前記イントラ符号化フレーム、前記前方向予測符号化フレーム、双方向予測符号化フレームの順番であり、かつ、前記双方向予測符号化フレーム内での順番は、符号化の順番とは逆であり、

前記出力手段は、前記復号手段により復号され、前記保持手段により保持されている前記双方向予測符号化フレームを順次読み出して出力するとともに、所定のタイミングで、前記保持手段により保持されている前記イントラ符号化フレーム、もしくは前記前方向予測符号化フレームを読み出して、前記双方向予測符号化フレームの間の所定の位置に挿入して出力することを特徴とする請求項9に記載の復号装置。

【請求項11】 前記所定の順番とは、前記出力手段により前記イントラ符号化フレーム、もしくは前記前方向予測符号化フレームが出力されるタイミングで、前記復号手段により復号された、一つ前の前記画像シーケンスの前記イントラ符号化フレームもしくは前記前方向予測符号化フレームが、前記保持手段により保持されるような順番であることを特徴とする請求項10に記載の復号装置。

【請求項12】 前記符号化ストリームを復号するため

に必要な情報を記録する記録手段と、
前記記録手段による前記情報の記録および前記情報の前記復号手段への供給を制御する制御手段とを更に備え、
前記符号化ストリームは、前記情報を含み、
前記制御手段は、前記復号手段の復号処理に必要な前記情報を選択して、前記復号手段に供給することを特徴とする請求項9に記載の復号装置。

【請求項13】 前記制御手段が前記復号手段に供給する前記情報は、前記復号手段により復号されているフレームに対応する上位層符号化パラメータであることを特徴とする請求項12に記載の復号装置。

【請求項14】 前記保持手段により保持されている前記画像データを読み出して出力する出力手段を更に備え、

前記復号手段は、前記符号化ストリームを通常再生に必要な処理速度のN倍速で復号することが可能であり、
前記出力手段は、前記保持手段により保持されている前記画像データのうちNフレーム毎の前記画像データを出力することが可能であることを特徴とする請求項7に記載の再生装置。

【請求項15】 前記符号化ストリームを保持する第1の保持手段と、

前記符号化ストリームから、前記符号化ストリームに含まれる所定の情報の単位の始まりを表わすスタートコードを読み出すとともに、前記第1の保持手段に前記スタートコードが保持されている位置に関する位置情報を読み出す読み出し手段と、

前記読み出し手段により読み出された、前記スタートコードおよび前記位置情報を保持する第2の保持手段と、
前記第1の保持手段による前記画像データの保持、および前記第2の保持手段による前記スタートコードおよび前記位置情報の保持を制御する第1の保持制御手段と、
複数の前記復号手段により復号され、出力された複数の画像データのうちの所定のものを選択する選択手段と、
前記選択手段により選択された前記画像データの入力を受け、必要に応じて動き補償を施す動き補償手段と、
前記選択手段により選択された前記画像データ、または前記動き補償手段により動き補償が施された前記画像データを保持する第3の保持手段と、

前記選択手段により選択された前記画像データ、または前記動き補償手段により動き補償が施された前記画像データの前記第3の保持手段による保持を、前記第1の保持制御手段とは独立して制御する第2の保持制御手段とを更に備えることを特徴とする請求項1に記載の復号装置。

【請求項16】 符号化ストリームを復号する復号装置の復号方法において、

前記符号化ストリームを復号する複数の復号ステップと、

複数の前記復号ステップの処理を並行して動作させるよ

うに制御する復号制御ステップとを含むことを特徴とする復号方法。

【請求項17】 符号化ストリームを復号する復号装置用のプログラムであって、

前記符号化ストリームを復号する複数の復号ステップと、
複数の前記復号ステップの処理を並行して動作させるように制御する復号制御ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項18】 符号化ストリームを復号する復号装置を制御するコンピュータが実行可能なプログラムであって、

前記符号化ストリームを復号する複数の復号ステップと、
複数の前記復号ステップの処理を並行して動作させるように制御する復号制御ステップとを含むことを特徴とするプログラム。

【請求項19】 符号化ストリームを復号する復号装置において、

前記符号化ストリームを復号する複数のスライスデコーダと、
複数の前記スライスデコーダを並行して動作させるように制御するスライスデコーダ制御手段とを備えることを特徴とする復号装置。

【請求項20】 符号化ストリームを復号する復号装置の復号方法において、

前記符号化ストリームを復号する複数のスライスデコーダによる復号を制御する復号制御ステップと、
前記復号制御ステップを並行して処理させるように制御するスライスデコーダ制御ステップとを含むことを特徴とする復号方法。

【請求項21】 符号化ストリームを復号する復号装置用のプログラムであって、

前記符号化ストリームを復号する複数のスライスデコーダによる復号を制御する復号制御ステップと、
前記復号制御ステップを並行して処理させるように制御するスライスデコーダ制御ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項22】 符号化ストリームを復号する復号装置を制御するコンピュータが実行可能なプログラムであって、

前記符号化ストリームを復号する複数のスライスデコーダによる復号を制御する復号制御ステップと、
前記復号制御ステップを並行して処理させるように制御するスライスデコーダ制御ステップとを含むことを特徴とするプログラム。

【請求項23】 ソース符号化ストリームを復号する復号装置において、

前記ソース符号化ストリームのピクチャを構成するスライス毎に、前記ソース符号化ストリームをデコードする複数のスライスデコーダと、

前記複数のスライスデコーダのデコードステータスを監視するとともに、前記複数のスライスデコーダを制御する制御手段とを備え、

前記制御手段は、前記ピクチャに含まれる前記スライスの順番に関係なく、前記スライスデコーダによる前記ピクチャのデコード処理が最も早くなるように、前記スライスを前記複数のスライスデコーダに割り当てることを特徴とする符号化装置。

【請求項24】 ソース符号化ストリームを復号する復号装置の復号方法において、

複数のスライスデコーダによる、前記ソース符号化ストリームのピクチャを構成するスライス毎の前記ソース符号化ストリームのデコード処理を制御するデコード処理制御ステップと、

前記複数のスライスデコーダのデコードステータスを監視するとともに、前記複数のスライスデコーダを制御する制御ステップとを含み、

前記制御ステップの処理では、前記ピクチャに含まれる前記スライスの順番に関係なく、前記スライスデコーダにおいて実行される前記デコード処理が最も早くなるように、前記スライスを前記複数のスライスデコーダに割り当てることを特徴とする符号化方法。

【請求項25】 ソース符号化ストリームを復号する復号装置を制御するコンピュータが実行可能なプログラムであって、

複数のスライスデコーダによる、前記ソース符号化ストリームのピクチャを構成するスライス毎の前記ソース符号化ストリームのデコード処理を制御するデコード処理制御ステップと、

前記複数のスライスデコーダのデコードステータスを監視するとともに、前記複数のスライスデコーダを制御する制御ステップとを含み、

前記制御ステップの処理では、前記ピクチャに含まれる前記スライスの順番に関係なく、前記スライスデコーダにおいて実行される前記デコード処理が最も早くなるように、前記スライスを前記複数のスライスデコーダに割り当てることを特徴とするプログラム。

【請求項26】 ソース符号化ストリームを復号する復号装置において、

前記ソース符号化ストリームのピクチャを構成するスライス毎に、前記ソース符号化ストリームをデコードする複数のスライスデコーダと、

前記複数のスライスデコーダのデコードステータスを監視するとともに、前記複数のスライスデコーダを制御する制御手段とを備え、

前記制御手段は、前記ピクチャに含まれる前記スライスの順番に関係なく、前記複数のスライスデコーダのうち

デコードが終了した前記スライスデコーダに、デコードすべき前記スライスを割り当てることを特徴とする符号化装置。

【請求項27】 ソース符号化ストリームを復号する復号装置の復号方法において、

複数のスライスデコーダによる、前記ソース符号化ストリームのピクチャを構成するスライス毎の前記ソース符号化ストリームのデコード処理を制御するデコード処理制御ステップと、

前記複数のスライスデコーダのデコードステータスを監視するとともに、前記複数のスライスデコーダを制御する制御ステップとを含み、

前記制御ステップの処理では、前記ピクチャに含まれる前記スライスの順番に関係なく、前記複数のスライスデコーダのうち、前記デコード処理制御ステップの処理によりデコード処理が終了した前記スライスデコーダに、デコードすべき前記スライスを割り当てることを特徴とする符号化方法。

【請求項28】 ソース符号化ストリームを復号する復号装置を制御するコンピュータが実行可能なプログラムであって、

複数のスライスデコーダによる、前記ソース符号化ストリームのピクチャを構成するスライス毎の前記ソース符号化ストリームのデコード処理を制御するデコード処理制御ステップと、

前記複数のスライスデコーダのデコードステータスを監視するとともに、前記複数のスライスデコーダを制御する制御ステップとを含み、

前記制御ステップの処理では、前記ピクチャに含まれる前記スライスの順番に関係なく、前記複数のスライスデコーダのうち、前記デコード処理制御ステップの処理によりデコード処理が終了した前記スライスデコーダに、デコードすべき前記スライスを割り当てることを特徴とするプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、復号装置および復号方法、並びに記録媒体に関し、特に、実現可能な回路規模で実時間動作が可能な4:2:2 P@HLに対応したビデオデコーダを実現することができる復号装置および復号方法、並びに記録媒体に関する。

【0002】

【従来の技術】MPEG2 (Moving Picture Coding Experts Group / Moving Picture Experts Group2) ビデオは、ISO/IEC (International Standards Organization / International Electrotechnical Commission) 13818-2、およびITU-T (International Telecommunication Union-Telecommunication sector) 勧告H.262に規定されているビデオ信号の高効率符号化方式である。

【0003】MPEG2の符号化ストリームは、符号化の手

法によって決まるプロファイルと、取り扱う画素数によって決まるレベルによってクラス分けされ、広範囲なアプリケーションに対応できるようになされている。例えば、MP@ML (メイン・プロファイル・メイン・レベル) は、そのクラスの1つであり、DVB (Digital Video Broadcast) や、DVD (Digital Versatile Disk) に広く実用化されている。プロファイルおよびレベルは、図5を用いて後述するsequence_extensionに記述される。

【0004】また、放送局におけるビデオ信号の制作には、ビデオの色差信号を、従来のベースバンドと同様の4:2:2方式で取り扱い、ビットレートの上限を増加した4:2:2P (4:2:2プロファイル) が規定されている。更に、次世代の高解像度ビデオ信号に対応するために、HL (ハイ・レベル) が規定されている。

【0005】図1に、MPEG2の代表的なクラスと、それぞれのクラスにおける、各種パラメータの上限値を示す。図1には、4:2:2P@HL (4:2:2プロファイル・ハイ・レベル)、4:2:2P@ML (4:2:2プロファイル・メイン・レベル)、MP@HL (メイン・プロファイル・ハイ・レベル)、MP@HL-1440 (メイン・プロファイル・ハイ・レベル-1440)、MP@ML (メイン・プロファイル・メイン・レベル)、MP@LL (メイン・プロファイル・ロー・レベル)、および、SP@ML (シンプル・プロファイル・メイン・レベル) について、ビットレート、1ラインあたりのサンプル数、1フレームあたりのライン数、フレーム周波数、およびサンプルの処理時間の上限値がそれぞれ示されている。

【0006】図1から、4:2:2P@HLのビットレートの上限値は、300 (Mbit/sec) であり、処理する画素数の上限値は、62,668,800 (samples/sec) である。一方、MP@MPのビットレートの上限値は、15 (Mbit/sec) であり、処理する画素数の上限値は、10,368,000 (samples/sec) である。すなわち、4:2:2P@HLをデコードするビデオデコーダは、MP@MLをデコードするビデオデコーダと比較して、ビットレートで20倍、処理する画素数で約6倍の処理能力が必要であることがわかる。

【0007】図2に、MPEG2ビデオビットストリームのレベル構造を示す。

【0008】最上位層であるピクチャ層の最初には、sequence_headerが記述されている。sequence_headerは、MPEGビットストリームのシーケンスのヘッダデータを定義するものである。シーケンス最初のsequence_headerに、sequence_extensionが続かない場合、このビットストリームには、ISO/IEC11172-2の規定が適応される。シーケンスの最初のsequence_headerに、sequence_extensionが続く場合、その後発生する全てのsequence_headerの直後には、sequence_extensionが続く。すなわち、図2に示す場合においては、全てのsequence_headerの直後に、sequence_extensionが続く。

【0009】sequence_extensionは、MPEGビットストリームのシーケンス層の拡張データを定義するものである。sequence_extensionは、sequence_headerの直後のみ発生し、かつ、復号後、およびフレームリオーダーリング後にフレームの損失がないようにするために、ビットストリームの最後に位置するsequence_end_codeの直前にきてはならない。また、ビットストリーム中に、sequence_extensionが発生した場合、それぞれのpicture_headerの直後にpicture_coding_extensionが続く。

【0010】GOP (group_of_picture) 内には、複数の画像 (picture) が含まれる。GOP_headerは、MPEGビットストリームのGOP層のヘッダデータを定義するものであり、更に、このビットストリーム中には、picture_headerとpicture_coding_extensionによって定義されたデータエレメントが記述されている。1枚の画像は、picture_headerおよびpicture_coding_extensionに続くpicture_dataとして符号化される。また、GOP_headerに続く最初の符号化フレームは、符号化されたIフレームである (すなわち、GOP_headerの最初の画像はIピクチャである)。ITU-T勧告H.262には、sequence_extensionおよびpicture_coding_extensionの他、各種の拡張が定義されているが、ここでは図示、および説明は省略する。

【0011】picture_headerは、MPEGビットストリームのピクチャ層のヘッダデータを定義するものであり、picture_coding_extensionは、MPEGビットストリームのピクチャ層の拡張データを定義するものである。

【0012】picture_dataは、MPEGビットストリームのスライス層およびマクロブロック層に関するデータエレメントを記述するものである。picture_dataは、図2に示されるように、複数のslice (スライス) に分割され、スライスは、複数のmacro_block (マクロブロック) に分割される。

【0013】macro_blockは、16×16の画素データで構成されている。スライスの最初のマクロブロックおよび最後のマクロブロックは、スキップマクロブロック (情報を含まないマクロブロック) ではない。マクロブロックは、16×16の画素データで構成されている。また、各ブロックは、8×8の画素データで構成されている。また、フレームDCT (Discrete Cosine Transform: 離散コサイン変換) 符号化およびフィールドDCT符号化の使用が可能なフレーム画像においては、フレーム符号化とフィールド符号化の場合で、マクロブロックの内部構成が相違する。

【0014】マクロブロックは、輝度成分および色差成分の1区画を含む。マクロブロックという用語は、情報源および復号データまたは対応する符号化データ成分のいずれかを示す。マクロブロックには、4:2:0、4:2:2および4:4:4の3つの色差フォーマットがある。マクロブロックにおけるブロックの順序は、そ

れぞれの色差フォーマットによって異なる。

【0015】図3(A)に、4:2:0方式の場合におけるマクロブロックを示す。4:2:0方式の場合、マクロブロックは、4個の輝度(Y)ブロックと、それぞれ1個の色差(Cb, Cr)ブロックで構成される。図3(B)に、4:2:2方式の場合におけるマクロブロックを示す。4:2:2方式の場合、マクロブロックは、4個の輝度(Y)ブロックと、それぞれ2個の色差(Cb, Cr)ブロックで構成される。

【0016】各マクロブロックは、いくつかの方法により、予測符号化処理が可能である。予測モードは、フィールド予測とフレーム予測の2種類に大別される。フィールド予測においては、先に復号された、1つ、もしくは複数のフィールドのデータを使用し、各フィールドについて、独自に予測を行う。フレーム予測は、先に復号された、1つ、もしくは複数のフレームを使用してフレームの予測を行う。フィールド画像内では、予測は全てフィールド予測である。一方、フレーム画像においては、フィールド予測、またはフレーム予測のいずれかにより予測が可能であり、その予測方法は、マクロブロックごとに選択される。また、マクロブロックの予測符号化処理においては、フィールド予測およびフレーム予測以外に、16×8動き補償およびデュアルプライムの2種類の特別予測モードを使用することができる。

【0017】動きベクトル情報、および他の周辺情報は、各マクロブロックの予測誤差信号とともに符号化される。動きベクトルの符号化については、可変長符号を使用して符号化された最後の動きベクトルを予測ベクトルとして、予測ベクトルとの差分ベクトルを符号化する。表示可能なベクトルの最大長は、画像毎にプログラムすることができ、適切な動きベクトルの計算は、符号器が行う。

【0018】そして、picture_dataの次には、sequence_headerとsequence_extensionが配置されている。このsequence_headerとsequence_extensionによって記述されたデータエレメントは、ビデオストリームのシーケンスの先頭に記述されたsequence_headerとsequence_extensionによって記述されたデータエレメントと全く同じである。このように同じデータをストリーム中に記述する理由は、ビットストリーム受信装置側でデータストリームの途中(例えばピクチャ層に対応するビットストリーム部分)から受信が開始された場合に、シーケンス層のデータを受信できなくなり、ストリームをデコード出来なくなることを防止するためである。

【0019】最後のsequence_headerとsequence_extensionとによって定義されたデータエレメントの次、つまり、データストリームの最後には、シーケンスの終わりを示す32ビットのsequence_end_codeが記述されている。

【0020】次に、図4乃至12を用いて、それぞれの

データエレメントの詳細について説明する。

【0021】図4に、sequence_headerのデータ構成を示す。sequence_headerに含まれるデータエレメントは、sequence_header_code、horizontal_size_value、vertical_size_value、aspect_ratio_information、frame_rate_code、bit_rate_value、marker_bit、vbv_buffer_size_value、constrained_parameter_flag、load_intra_quantiser_matrix、intra_quantiser_matrix、load_non_intra_quantiser_matrix、およびnon_intra_quantiser_matrix等から構成される。

【0022】sequence_header_codeは、シーケンス層のスタート同期コードを表すデータである。horizontal_size_valueは、画像の水平方向の画素数の下位12ビットからなるデータである。vertical_size_valueは、画像の縦のライン数の下位12ビットからなるデータである。aspect_ratio_informationは、画素のアスペクト比(縦横比)または表示画面アスペクト比を表すデータである。frame_rate_codeは、画像の表示周期を表すデータである。bit_rate_valueは、発生ビット量に対する制限のためのビットレートの下位18ビットのデータである。

【0023】そして、marker_bitは、スタートコードエミュレーションを防止するために挿入されるビットデータである。vbv_buffer_size_valueは、発生符号量制御用の仮想バッファV B V (Video Buffering Verifier)の大きさを決める値の下位10ビットデータである。constrained_parameter_flagは、各パラメータが制限以内であることを示すデータである。load_non_intra_quantiser_matrixは、非イントラMB用量子化マトリックス・データの存在を示すデータである。load_intra_quantiser_matrixは、イントラMB用量子化マトリックス・データの存在を示すデータである。intra_quantiser_matrixは、イントラMB用量子化マトリックスの値を示すデータである。non_intra_quantiser_matrixは、非イントラMB用量子化マトリックスの値を表すデータである。

【0024】図5に、sequence_extensionのデータ構成を示す。sequence_extensionは、extension_start_code、extension_start_code_identifier、profile_and_level_indication、progressive_sequence、chroma_format、horizontal_size_extension、vertical_size_extension、bit_rate_extension、marker_bit、vbv_buffer_size_extension、low_delay、frame_rate_extension_n、およびframe_rate_extension_d等のデータエレメントから構成されている。

【0025】extension_start_codeは、エクステンションデータのスタート同期コードを表すデータである。extension_start_code_identifierは、どの拡張データが送られるかを示すデータである。profile_and_level_indicationは、ビデオデータのプロファイルとレベルを指定するためのデータである。progressive_sequenceは、

ビデオデータが順次走査（プログレッシブ画像）であることを示すデータである。chroma_formatは、ビデオデータの色差フォーマットを指定するためのデータである。horizontal_size_extensionは、シーケンスヘッダのhorizontal_size_valueに加える上位2ビットのデータである。vertical_size_extensionは、シーケンスヘッダのvertical_size_valueに加える上位2ビットのデータである。

【0026】そして、bit_rate_extensionは、シーケンスヘッダのbit_rate_valueに加える上位12ビットのデータである。marker_bitは、スタートコードエミュレーションを防止するために挿入されるビットデータである。vbv_buffer_size_extensionは、シーケンスヘッダのvbv_buffer_size_valueに加える上位8ビットのデータである。low_delayは、Bピクチャを含まないことを示すデータである。frame_rate_extension_nは、シーケンスヘッダのframe_rate_codeと組み合わせてフレームレートを取得するためのデータである。frame_rate_extension_dは、シーケンスヘッダのframe_rate_codeと組み合わせてフレームレートを取得するためのデータである。

【0027】図6に、GOP_headerのデータ構成を示す。GOP_headerを表わすデータエレメントは、group_start_code、time_code、closed_gop、およびbroken_linkから構成される。

【0028】group_start_codeは、GOP層の開始同期コードを示すデータである。time_codeは、GOPの先頭ピクチャの時間を示すタイムコードである。closed_gopは、GOP内の画像が他のGOPから独立再生可能なことを示すフラグデータである。broken_linkは、編集などのためにGOP内の先頭のBピクチャが正確に再生できないことを示すフラグデータである。

【0029】図7に、picture_headerのデータ構成を示す。picture_headerに関するデータエレメントは、picture_start_code、temporal_reference、picture_coding_type、vbv_delay、full_pel_forward_vector、forward_f_code、full_pel_backward_vector、およびbackward_f_code等から構成される。

【0030】picture_start_codeは、ピクチャ層の開始同期コードを表すデータである。temporal_referenceは、ピクチャの表示順を示す番号でGOPの先頭でリセットされるデータである。picture_coding_typeは、ピクチャタイプを示すデータである。vbv_delayは、ランダムアクセス時の仮想バッファの初期状態を示すデータである。full_pel_forward_vector、forward_f_code、full_pel_backward_vector、およびbackward_f_codeは、MP EG2では使用されない固定データである。

【0031】図8に、picture_coding_extensionのデータ構成を示す。picture_coding_extensionは、extension_start_code、extension_start_code_identifier、f_code[0][0]、f_code[0][1]、f_code[1][0]、f_code[1]

[1]、intra_dc_precision、picture_structure、top_field_first、frame_pred_frame_dct、concealment_motion_vectors、q_scale_type、intra_vlc_format、alternate_scan、repeat_firt_field、chroma_420_type、progressive_frame、composite_display_flag、v_axis、field_sequence、sub_carrier、burst_amplitude、およびsub_carrier_phase等から構成される。

【0032】extension_start_codeは、ピクチャ層のエクステンションデータのスタートを示す開始コードである。extension_start_code_identifierは、どの拡張データが送られるかを示すコードである。f_code[0][0]は、フォワード方向の水平動きベクトル探索範囲を表すデータである。f_code[0][1]は、フォワード方向の垂直動きベクトル探索範囲を表すデータである。f_code[1][0]は、バックワード方向の水平動きベクトル探索範囲を表すデータである。f_code[1][1]は、バックワード方向の垂直動きベクトル探索範囲を表すデータである。

【0033】intra_dc_precisionは、DC係数の精度を表すデータである。ブロック内の各画素の輝度および色差信号を表した行列fに、DCTを施すと、8×8のDCT係数行列Fが得られる。この行列Fの左上隅の係数をDC係数と呼ぶ。DC係数はブロック内の平均輝度、平均色差を表わす信号である。picture_structureは、フレームストラクチャかフィールドストラクチャかを示すデータであり、フィールドストラクチャの場合は、上位フィールドか下位フィールドかもあわせて示すデータである。top_field_firstは、フレームストラクチャの場合、最初のフィールドが上位か下位かを示すデータである。frame_predictive_frame_dctは、フレームストラクチャの場合、フレーム・モードDCTの予測がフレーム・モードだけであることを示すデータである。concealment_motion_vectorsは、イントラマクロブロックに伝送エラーを隠蔽するための動きベクトルがついていることを示すデータである。

【0034】q_scale_typeは、線形量子化スケールを利用するか、非線形量子化スケールを利用するかを示すデータである。intra_vlc_formatは、イントラマクロブロックに、別の2次元VLC（Variable Length Coding）を使うか否かを示すデータである。alternate_scanは、ジグザグスキャンを使うか、オルタネート・スキャンを使うかの選択を表すデータである。repeat_firt_fieldは、2：3プルダウンの際に使われるデータである。chroma_420_typeは、信号フォーマットが4：2：0の場合、次のprogressive_frameと同じ値であり、そうでない場合は0を表すデータである。progressive_frameは、このピクチャが、順次走査かインタレースフィールドかを示すデータである。composite_display_flagは、ソース信号がコンポジット信号であったかどうかを示すデータである。v_axis、field_sequence、sub_carrier、burst_amplitude、およびsub_carrier_phaseは、ソ

ース信号がコンポジット信号の場合に使われるデータである。

【0035】図9に、picture_dataのデータ構成を示す。picture_data () 関数によって定義されるデータエレメントは、slice () 関数によって定義されるデータエレメントである。このslice () 関数によって定義されるデータエレメントは、ビットストリーム中に少なくとも1個記述されている。

【0036】slice () 関数は、図10に示されるように、slice_start_code、quantiser_scale_code、intra_slice_flag、intra_slice、reserved_bits、extra_bit_slice、およびextra_information_slice等のデータエレメントと、macroblock () 関数によって定義される。

【0037】slice_start_codeは、slice () 関数によって定義されるデータエレメントのスタートを示すスタートコードである。quantiser_scale_codeは、このスライス層に存在するマクロブロックに対して設定された量子化ステップサイズを示すデータであるが、マクロブロック毎に、quantiser_scale_codeが設定されている場合には、各マクロブロックに対して設定されたmacroblock_quantiser_scale_codeのデータが優先して使用される。

【0038】intra_slice_flagは、ビットストリーム中にintra_sliceおよびreserved_bitsが存在するか否かを示すフラグである。intra_sliceは、スライス層中にノンイントラマクロブロックが存在するか否かを示すデータである。スライス層におけるマクロブロックのいずれかがノンイントラマクロブロックである場合には、intra_sliceは「0」となり、スライス層におけるマクロブロックの全てがノンイントラマクロブロックである場合には、intra_sliceは「1」となる。reserved_bitsは、7ビットのデータであって「0」の値を取る。extra_bit_sliceは、追加の情報が存在することを示すフラグであって、次にextra_information_sliceが存在する場合には「1」に設定され、追加の情報が存在しない場合には「0」に設定される。

【0039】これらのデータエレメントの次には、macroblock () 関数によって定義されたデータエレメントが記述されている。macroblock () 関数は、図11に示すように、macroblock_escape、macroblock_address_increment、およびquantiser_scale_code、およびmarker_bit等のデータエレメントと、macroblock_modes () 関数、motion_vectors (s) 関数、およびcoded_block_pattern () 関数によって定義されたデータエレメントを記述するための関数である。

【0040】macroblock_escapeは、参照マクロブロックと前のマクロブロックとの水平方向の差が34以上であるか否かを示す固定ビット列である。参照マクロブロックと前のマクロブロックとの水平方向の差が34以上の場合には、macroblock_address_incrementの値に33

が加えられる。macroblock_address_incrementは、参照マクロブロックと前のマクロブロックとの水平方向の差を示すデータである。もし、macroblock_address_incrementの前にmacroblock_escapeが1つ存在するのであれば、このmacroblock_address_incrementの値に33を加えた値が、実際の参照マクロブロックと前のマクロブロックとの水平方向の差分を示すデータとなる。

【0041】quantiser_scale_codeは、各マクロブロックに設定された量子化ステップサイズを示すデータであり、macroblock_quantが「1」のときだけ存在する。各スライス層には、スライス層の量子化ステップサイズを示すslice_quantiser_scale_codeが設定されているが、参照マクロブロックに対してscale_codeが設定されている場合には、この量子化ステップサイズを選択する。

【0042】macroblock_address_incrementの次には、macroblock_modes () 関数によって定義されるデータエレメントが記述されている。macroblock_modes () 関数は、図12に示すように、macroblock_type、frame_motion_type、field_motion_type、dct_type等のデータエレメントを記述するための関数である。macroblock_typeは、マクロブロックの符号化タイプを示すデータである。

【0043】macroblock_motion_forward又はmacroblock_motion_backwardが「1」であり、ピクチャ構造がフレームであり、更にframe_pred_frame_dctが「0」である場合には、macroblock_typeを表わすデータエレメントの次にframe_motion_typeを表わすデータエレメントが記述されている。なお、このframe_pred_frame_dctは、frame_motion_typeがビットストリーム中に存在するか否かを示すフラグである。

【0044】frame_motion_typeは、フレームのマクロブロックの予測タイプを示す2ビットのコードである。予測ベクトルが2個で、フィールドベースの予測タイプであれば、frame_motion_typeは「00」であり、予測ベクトルが1個で、フィールドベースの予測タイプであれば、frame_motion_typeは「01」であり、予測ベクトルが1個で、フレームベースの予測タイプであれば、frame_motion_typeは「10」であり、予測ベクトルが1個で、デュアルプライムの予測タイプであれば、frame_motion_typeは「11」である。

【0045】field_motion_typeは、フィールドのマクロブロックの動き予測を示す2ビットのコードである。予測ベクトルが1個でフィールドベースの予測タイプであれば「01」であって、予測ベクトルが2個で18×8マクロブロックベースの予測タイプであれば「10」であって、予測ベクトルが1個でデュアルプライムの予測タイプであれば「11」である。

【0046】ピクチャ構造がフレームであり、frame_pred_frame_dctが、そのビットストリーム中にframe_motion_typeが存在することを示し、frame_pred_frame_dct

が、そのビットストリーム中にdct_typeが存在することを示している場合、macroblock_typeを表わすデータエレメントの次にはdct_typeを表わすデータエレメントが記述されている。なお、dct_typeは、DCTがフレームDCTモードか、フィールドDCTモードかを示すデータである。

【0047】MPEG2のストリームにおいて、以上説明したそれぞれのデータエレメントは、start codeと称される、特殊なビットパターンで開始される。これらのスタートコードは、別の状況では、ビデオストリーム中に現れない特定のビットパターンである。各スタートコードは、スタートコードプレフィクスと、それに続くスタートコード値から構成される。スタートコードプレフィクスは、ビット列“00000000 0000 0000 0000 0001”である。スタートコード値は、スタートコードのタイプを識別する8ビットの整数である。

【0048】図13に、MPEG2の各start codeの値を示す。多くのスタートコードは、1個のスタートコード値より示される。しかし、slice_start_codeは、01乃至AFの複数のスタートコード値により示され、このスタートコード値は、スライスに対する垂直位置を表わす。これらのスタートコードは、全てバイト単位であるため、スタートコードプレフィクスの最初のビットがバイトの最初のビットになるように、スタートコードプレフィクスの前に、複数のビット“0”が挿入され、スタートコードがバイト単位になるように調整される。

【0049】図14は、従来のMP@MLに対応したMPEGビデオデコーダの回路構成を示すブロック図である。

【0050】MPEGビデオデコーダは、ストリーム入力回路11、バッファ制御回路12、クロック発生回路13、スタートコード検出回路14、デコーダ15、動き補償回路16、および表示出力回路17から構成されるIC (integrated circuit) 1と、ストリームバッファ21およびビデオバッファ22で構成され、例えば、DRAM (Dynamic Random Access Memory) からなるバッファ2により構成される。

【0051】IC1のストリーム入力回路11は、高効率符号化された符号化ストリームの入力を受け、バッファ制御回路12に供給する。バッファ制御回路12は、クロック発生回路13から供給される基本クロックに従って、入力された符号化ストリームをバッファ2のストリームバッファ21に入力する。ストリームバッファ21は、MP@MLのデコードに要求されるVBVバッファサイズである1,835,008ビットの容量を有する。ストリームバッファ21に保存されている符号化ストリームは、バッファ制御回路12の制御に従って、先に書き込まれたデータから順に読み出され、スタートコード検出回路14に供給される。スタートコード検出回路14は、入力されたストリームから、図13を用いて説明したスタートコードを検出し、検出したスタートコードお

よび入力されたストリームをデコーダ15に出力する。

【0052】デコーダ15は、入力されたストリームをMPEGシンタックスに基づいて、デコードする。デコーダ15は、入力されたスタートコードに従って、まず、ピクチャ層のヘッダパラメータをデコードし、それを基に、スライス層をマクロブロックに分離してマクロブロックをデコードし、その結果得られる予測ベクトルおよび画素を、動き補償回路16に出力する。

【0053】MPEGでは、画像の時間的冗長性を利用して、近接した画像間で動き補償した差分を得ることにより、符号化効率を改善している。MPEGビデオデコーダでは、動き補償を用いた画素に対しては、現在デコードしている画素にその動きベクトルが示す参照画像の画素データを加算することにより動き補償を行い、符号化前の画像データに復号する。

【0054】デコーダ15から出力されるマクロブロックが動き補償を使用していない場合、動き補償回路16は、その画素データを、バッファ制御回路12を介して、バッファ2のビデオバッファ22に書き込み、表示出力に備えるとともに、この画素データが、他の画像の参照データとされる場合に備える。

【0055】デコーダ15から出力されるマクロブロックが動き補償を使用している場合、動き補償回路16は、デコーダ15から出力される予測ベクトルに従って、バッファ制御回路12を介して、バッファ2のビデオバッファ22から参照画素データを読み出す。そして、読み出した参照画素データを、デコーダ15から供給された画素データに加算し、動き補償を行う。動き補償回路16は、動き補償を行った画素データを、バッファ制御回路12を介してバッファ2のビデオバッファ22に書き込み、表示出力に備えるとともに、この画素データが、他の画素の参照データとされる場合に備える。

【0056】表示出力回路17は、デコードした画像データを出力するための同期タイミング信号を発生し、このタイミングを基に、バッファ制御回路12を介して、ビデオバッファ22から画素データを読み出し、復号ビデオ信号として出力する。

【0057】

【発明が解決しようとする課題】以上説明したように、MPEG2ストリームは階層構造を有している。図2を用いて説明したピクチャ層のsequence_header乃至picture_coding_extensionのデータは、図1を用いて説明したプロファイルおよびレベルが異なる場合においても、そのデータ量は、あまり変更されない。一方、スライス層以下のデータ量は、符号化する画素数に依存する。

【0058】図1より、HLにおいて、1枚のピクチャで処理しなければならないマクロブロックの数は、MLに対して約6倍になる。更に、図3より、4:2:2Pにおいて、1個のマクロブロックで処理するブロックの数は、MPの4/3倍になる。

【0059】すなわち、図14を用いて説明したMP@MLに対応したビデオデコーダで、4:2:2 P@HLの符号化ストリームを復号しようとした場合、V B Vバッファサイズおよび画素数の増加に伴って、ストリームバッファ21のバッファサイズが不足する。また、ビットレートの増加に伴い、入力ストリームのストリームバッファ21へのアクセスが増加し、画素数の増加に伴って、動き補償回路16のビデオバッファ22へのアクセスが増加するため、バッファ制御回路12の制御が間に合わなくなる。更に、ビットレートの増加、マクロブロックおよびブロック数の増加に伴って、デコーダ15の処理が間に合わなくなる。

【0060】今日の半導体技術の進展により、信号処理回路、メモリ（バッファ）回路とも、その動作速度は著しく向上している。しかしながら、現在のML@MPの復号技術では、4:2:2 P@HLを復号するまでには至っていない。一般に、このような高速の信号処理を行おうとした場合、回路規模が大幅に増加し、部品点数の増加および消費電力の増加を招いてしまう。

【0061】本発明はこのような状況に鑑みてなされたものであり、今日の半導体技術を用いて、実現可能な回路規模で実時間動作が可能な4:2:2 P@HLに対応したビデオデコーダを実現することができるようにするのである。

【0062】

【課題を解決するための手段】本発明の第1の復号装置は、符号化ストリームを復号する複数の復号手段と、複数の復号手段を並行して動作させるように制御する復号制御手段とを備えることを特徴とする。

【0063】複数の復号手段には、復号処理の終了を示す信号を復号制御手段に出力させるようにすることができ、復号制御手段には、復号処理の終了を示す信号を出力した復号手段に、符号化ストリームを復号させるように制御させることができる。

【0064】符号化ストリームをバッファリングする第1のバッファ手段と、符号化ストリームから、符号化ストリームに含まれる所定の情報の単位の始まりを表わすスタートコードを読み出すとともに、第1のバッファ手段に、スタートコードが保持されている位置に関する位置情報を読み出す読み出し手段と、読み出し手段により読み出された、スタートコードおよび位置情報をバッファリングする第2のバッファ手段と、第1のバッファ手段による符号化ストリームのバッファリング、および第2のバッファ手段によるスタートコードおよび位置情報のバッファリングを制御するバッファリング制御手段とを更に備えさせるようにすることができる。

【0065】符号化ストリームは、ISO/IEC13818-2およびITU-T勧告H.262に規定されたMPEG2の符号化ストリームとすることができる。

【0066】複数の復号手段により復号され、出力され

た複数の画像データのうちの所定のものを選択する選択手段と、選択手段により選択された画像データの入力を受け、必要に応じて動き補償を施す動き補償手段とを更に備えさせるようにすることができる。

【0067】復号手段には、復号処理が終了したことを示す終了信号を選択手段に出力させるようにすることができ、選択手段には、複数の復号手段のそれぞれの処理状態に対応する値を記憶する記憶手段を有し、記憶手段の値が全て第1の値になった場合、復号処理が終了したことを示す終了信号を出力している復号手段に対応する記憶手段に記憶されている値を、第1の値から第2の値に変更させ、対応する記憶手段に記憶されている値が第2の値である復号手段により復号された画像データのうち、いずれかの画像データを選択させ、選択された画像データを復号した復号手段に対応する記憶手段に記憶されている値を第1の値に変更させるようにすることができる。

【0068】選択手段により選択された画像データ、または動き補償手段により動き補償が施された画像データを保持する保持手段と、選択手段により選択された画像データ、および動き補償手段により動き補償が施された画像データの保持手段による保持を制御する保持制御手段とを更に備えさせるようにすることができる。

【0069】保持手段には、画像データの輝度成分と色差成分をそれぞれ分けて保持させるようにすることができる。

【0070】復号手段に供給される符号化ストリームのフレームの順番を変更する変更手段を更に備えさせるようにことができ、保持手段には、画像シーケンス内のイントラ符号化フレームおよび前方向予測符号化フレームを合計したフレーム数より少なくとも2つ多い数のフレームを保持させるようにことができ、変更手段には、符号化ストリームを逆転再生させるための所定の順番になるように、符号化ストリームのフレームの順番を変更させるようにすることができる。

【0071】保持手段により保持されている画像データを読み出して出力する出力手段を更に備えさせるようにことができ、所定の順番とは、イントラ符号化フレーム、前方向予測符号化フレーム、双方向予測符号化フレームの順番であり、かつ、双方向予測符号化フレーム内の順番は、符号化の順番とは逆であるものとして、出力手段には、復号手段により復号され、保持手段により保持されている双方向予測符号化フレームを順次読み出して出力するとともに、所定のタイミングで、保持手段により保持されているイントラ符号化フレーム、もしくは前方向予測符号化フレームを読み出して、双方向予測符号化フレームの間の所定の位置に挿入して出力させるようにすることができる。

【0072】所定の順番とは、出力手段によりイントラ符号化フレーム、もしくは前方向予測符号化フレームが

出力されるタイミングで、復号手段により復号された、一つ前の画像シーケンスのイントラ符号化フレームもしくは前方向予測符号化フレームが、保持手段により保持されるような順番であるものとする事ができる。

【0073】符号化ストリームを復号するために必要な情報を記録する記録手段と、記録手段による情報の記録および情報の復号手段への供給を制御する制御手段とを更に備えさせるようにすることができ、符号化ストリームは、情報を含むものとする事ができ、制御手段には、復号手段の復号処理に必要な情報を選択して、復号手段に供給させるようにすることができ、

【0074】制御手段が復号手段に供給する情報は、復号手段により復号されているフレームに対応する上位層符号化パラメータであるものとする事ができる。

【0075】保持手段により保持されている画像データを読み出して出力する出力手段を更に備えさせるようにすることができ、復号手段は、符号化ストリームを通常再生に必要な処理速度のN倍速で復号することが可能であるものとする事ができ、出力手段は、保持手段により保持されている画像データのうちNフレーム毎の画像データを出力することが可能であるものとする事ができる。

【0076】符号化ストリームを保持する第1の保持手段と、符号化ストリームから、符号化ストリームに含まれる所定の情報の単位の始まりを表わすスタートコードを読み出すとともに、第1の保持手段に、スタートコードが保持されている位置に関する位置情報を読み出す読み出し手段と、読み出し手段により読み出された、スタートコードおよび位置情報を保持する第2の保持手段と、第1の保持手段による符号化ストリームの保持、および第2の保持手段によるスタートコードおよび位置情報の保持を制御する第1の保持制御手段と、複数の復号手段により復号され、出力された複数の画像データのうちの所定のものを選択する選択手段と、選択手段により選択された画像データの入力を受け、必要に応じて動き補償を施す動き補償手段と、選択手段により選択された画像データ、または動き補償手段により動き補償が施された画像データを保持する第3の保持手段と、選択手段により選択された画像データ、および動き補償手段により動き補償が施された画像データの第3の保持手段による保持を、第1の保持制御手段とは独立して制御する第2の保持制御手段とを更に備えさせることができる。

【0077】本発明の第1の復号方法は、符号化ストリームを復号する複数の復号ステップと、複数の復号ステップの処理を並行して動作させるように制御する復号制御ステップとを含むことを特徴とする。

【0078】本発明の第1の記録媒体に記録されているプログラムは、符号化ストリームを復号する複数の復号ステップと、複数の復号ステップの処理を並行して動作させるように制御する復号制御ステップとを含むことを

特徴とする。

【0079】本発明の第1のプログラムは、符号化ストリームを復号する複数の復号ステップと、複数の復号ステップの処理を並行して動作させるように制御する復号制御ステップとを含むことを特徴とする。

【0080】本発明の第2の復号装置は、符号化ストリームを復号する複数のスライスデコーダと、複数のスライスデコーダを並行して動作させるように制御するスライスデコーダ制御手段とを備えることを特徴とする。

【0081】本発明の第2の復号方法は、符号化ストリームを復号する複数のスライスデコーダによる復号を制御する復号制御ステップと、復号制御ステップを並行して処理させるように制御するスライスデコーダ制御ステップとを含むことを特徴とする。

【0082】本発明の第2の記録媒体に記録されているプログラムは、符号化ストリームを復号する複数のスライスデコーダによる復号を制御する復号制御ステップと、復号制御ステップを並行して処理させるように制御するスライスデコーダ制御ステップとを含むことを特徴とする。

【0083】本発明の第2のプログラムは、符号化ストリームを復号する複数のスライスデコーダによる復号を制御する復号制御ステップと、復号制御ステップを並行して処理させるように制御するスライスデコーダ制御ステップとを含むことを特徴とする。

【0084】本発明の第3の復号装置は、ソース符号化ストリームのピクチャを構成するスライス毎に、ソース符号化ストリームをデコードする複数のスライスデコーダと、複数のスライスデコーダのデコードステータスを監視するとともに、複数のスライスデコーダを制御する制御手段とを備え、制御手段は、ピクチャに含まれるスライスの順番に関係なく、スライスデコーダによるピクチャのデコード処理が最も早くなるように、スライスを複数のスライスデコーダに割り当てることを特徴とする。

【0085】本発明の第3の復号方法は、複数のスライスデコーダによる、ソース符号化ストリームのピクチャを構成するスライス毎のソース符号化ストリームのデコード処理を制御するデコード処理制御ステップと、複数のスライスデコーダのデコードステータスを監視するとともに、複数のスライスデコーダを制御する制御ステップとを含み、制御ステップの処理では、ピクチャに含まれるスライスの順番に関係なく、スライスデコーダにおいて実行されるデコード処理が最も早くなるように、スライスを複数のスライスデコーダに割り当てることを特徴とする。

【0086】本発明の第3のプログラムは、複数のスライスデコーダによる、ソース符号化ストリームのピクチャを構成するスライス毎のソース符号化ストリームのデコード処理を制御するデコード処理制御ステップと、複

数のスライスデコーダのデコードステータスを監視するとともに、複数のスライスデコーダを制御する制御ステップとを含み、制御ステップの処理では、ピクチャに含まれるスライスの順番に関係なく、スライスデコーダにおいて実行されるデコード処理が最も早くなるように、スライスを複数のスライスデコーダに割り当てることを特徴とする。

【0087】本発明の第4の復号装置は、ソース符号化ストリームのピクチャを構成するスライス毎に、ソース符号化ストリームをデコードする複数のスライスデコーダと、複数のスライスデコーダのデコードステータスを監視するとともに、複数のスライスデコーダを制御する制御手段とを備え、制御手段は、ピクチャに含まれるスライスの順番に関係なく、複数のスライスデコーダのうちデコードが終了したスライスデコーダに、デコードすべきスライスを割り当てることを特徴とする。

【0088】本発明の第4の復号方法は、複数のスライスデコーダによる、ソース符号化ストリームのピクチャを構成するスライス毎のソース符号化ストリームのデコード処理を制御するデコード処理制御ステップと、複数のスライスデコーダのデコードステータスを監視するとともに、複数のスライスデコーダを制御する制御ステップとを含み、制御ステップの処理では、ピクチャに含まれるスライスの順番に関係なく、複数のスライスデコーダのうち、デコード処理制御ステップの処理によりデコード処理が終了したスライスデコーダに、デコードすべきスライスを割り当てることを特徴とする。

【0089】本発明の第4のプログラムは、複数のスライスデコーダによる、ソース符号化ストリームのピクチャを構成するスライス毎のソース符号化ストリームのデコード処理を制御するデコード処理制御ステップと、複数のスライスデコーダのデコードステータスを監視するとともに、複数のスライスデコーダを制御する制御ステップとを含み、制御ステップの処理では、ピクチャに含まれるスライスの順番に関係なく、複数のスライスデコーダのうち、デコード処理制御ステップの処理によりデコード処理が終了したスライスデコーダに、デコードすべきスライスを割り当てることを特徴とする。

【0090】本発明の第1の復号装置、復号方法、およびプログラムにおいては、符号化ストリームが復号され、復号処理を並行して動作させるように、復号処理が制御される。

【0091】本発明の第2の復号装置、復号方法、およびプログラムにおいては、符号化ストリームが複数のスライスデコーダによって復号され、複数のスライスデコーダによる復号処理が並行して行われる。

【0092】本発明の第3の復号装置、復号方法、およびプログラムにおいては、ソース符号化ストリームのピクチャを構成するスライス毎に、ソース符号化ストリームがデコードされ、複数のスライスデコーダのデコード

ステータスが監視されるとともに、複数のスライスデコーダが制御され、ピクチャに含まれるスライスの順番に関係なく、スライスデコーダにおいて実行されるデコード処理が最も早くなるように、スライスが複数のスライスデコーダに割り当てられる。

【0093】本発明の第4の復号装置、復号方法、およびプログラムにおいては、ソース符号化ストリームのピクチャを構成するスライス毎に、ソース符号化ストリームがデコードされ、複数のスライスデコーダのデコードステータスが監視されるとともに、複数のスライスデコーダが制御され、ピクチャに含まれるスライスの順番に関係なく、複数のスライスデコーダのうちデコードが終了したスライスデコーダに、デコードすべきスライスが割り当てられる。

【0094】

【発明の実施の形態】以下、図を参照して、本発明の実施の形態について説明する。

【0095】図15は、本発明を適応した、MPEGビデオデコーダの回路構成を示すブロック図である。

【0096】図15のMPEGビデオデコーダは、ストリーム入力回路41、スタートコード検出回路42、ストリームバッファ制御回路43、クロック発生回路44、ピクチャデコーダ45、スライスデコーダ制御回路46、スライスデコーダ47乃至49、動き補償回路50、輝度バッファ制御回路51、色差バッファ制御回路52、および表示出力回路53から構成されるIC31、ストリームバッファ61およびスタートコードバッファ62で構成され、例えば、DRAMからなるバッファ32、輝度バッファ71および色差バッファ72で構成され、例えば、DRAMからなるビデオバッファ33、コントローラ34、並びに、ドライブ35で構成される。

【0097】ストリーム入力回路41は、高能率符号化された符号化ストリームの入力を受け、スタートコード検出回路42に供給する。スタートコード検出回路42は、入力された符号化ストリームをストリームバッファ制御回路43に供給するとともに、図13を用いて説明したスタートコードを検出して、それを基に、そのスタートコードの種類と、ストリームバッファ61にそのスタートコードが書き込まれる位置を示す書き込みポイントを含む、スタートコード情報を生成し、ストリームバッファ制御回路43に供給する。

【0098】クロック発生回路44は、図14を用いて説明したクロック発生回路13の2倍の基本クロックを発生し、ストリームバッファ制御回路43に供給する。ストリームバッファ制御回路43は、クロック発生回路44から供給される基本クロックに従って、入力された符号化ストリームを、バッファ32のストリームバッファ61に書き込み、入力されたスタートコード情報を、バッファ32のスタートコードバッファ62に書き込む。

【0099】MPEGビデオデコーダが、4:2:2 P@HLのMPEG符号化ストリームを順方向に再生することができるようになされている場合、ストリームバッファ61は、少なくとも、4:2:2 P@HLのデコードに要求されるV B Vバッファサイズである47,185,920ビットの容量を有している。また、MPEGビデオデコーダが、逆転再生を実行することができるようになされている場合、ストリームバッファ61は、少なくとも、2 GOP分のデータを記録することができる容量を有している。

【0100】ピクチャデコーダ45は、ストリームバッファ制御回路43を介して、スタートコードバッファ62からスタートコード情報を読み出す。例えば、デコード開始時は、図2を用いて説明したsequence_headerからデコードが開始されるので、ピクチャデコーダ45は、図4を用いて説明したスタートコードであるsequence_header_codeに対応する書き込みポインタをスタートコードバッファ62から読み出し、その書き込みポインタを基に、ストリームバッファ61からsequence_headerを読み出してデコードする。続いて、ピクチャデコーダ45は、sequence_headerの読み出しと同様に、sequence_extension、GOP_header、picture_coding_extension等をストリームバッファ61から読み出してデコードする。

【0101】ピクチャデコーダ45が、スタートコードバッファ62から、最初のslice_start_codeを読み出した時点で、そのピクチャのデコードに必要な全てのパラメータが揃ったことになる。ピクチャデコーダ45は、デコードしたピクチャ層のパラメータを、スライスデコーダ制御回路46に出力する。

【0102】スライスデコーダ制御回路46は、ピクチャ層のパラメータの入力を受け、ストリームバッファ制御回路43を介して、スタートコードバッファ62から、対応するスライスのスタートコード情報を読み出す。また、スライスデコーダ制御回路46は、スライスデコーダ47乃至49のいずれかにデコードさせるスライスが、符号化ストリームに含まれる何番目のスライスであるかを示すレジスタを有し、そのレジスタを参照しながら、ピクチャ層のパラメータと、スタートコード情報に含まれるスライスの書き込みポインタをスライスデコーダ47乃至49のいずれかに供給する。スライスデコーダ制御回路46が、スライスデコーダ47乃至49のうち、デコードを実行させるスライスデコーダを選択する処理については、図16および図17を用いて後述する。

【0103】スライスデコーダ47は、マクロブロック検出回路81、ベクトル復号回路82、逆量子化回路83、および逆DCT回路84で構成され、スライスデコーダ制御回路46から入力されたスライスの書き込みポインタを基に、対応するスライスを、ストリームバッファ制御回路43を介してストリームバッファ61から読

み出す。そして、スライスデコーダ制御回路46から入力されたピクチャ層のパラメータに従って、読み出したスライスをデコードして、動き補償回路50に出力する。

【0104】マクロブロック検出回路81は、スライス層のマクロブロックを分離し、各マクロブロックのパラメータをデコードし、可変長符号化された各マクロブロックの予測モードおよび予測ベクトルをベクトル復号回路82に供給し、可変長符号化された係数データを逆量子化回路83に供給する。ベクトル復号回路82は、可変長符号化された、各マクロブロックの予測モードおよび予測ベクトルをデコードして、予測ベクトルを復元する。逆量子化回路83は、可変長符号化された係数データをデコードして逆DCT回路84に供給する。逆DCT回路84は、デコードされた係数データに逆DCTを施し、符号化前の画素データに復元する。

【0105】スライスデコーダ47は、動き補償回路50に、デコードしたマクロブロックに対する動き補償の実行を要求し（すなわち、図中、REQで示される信号を1にする）、動き補償回路50から動き補償の実行要求に対する受付を示す信号（図中ACKで示される信号）を受けて、デコードされた予測ベクトルおよびデコードされた画素を動き補償回路50に供給する。スライスデコーダ47は、ACK信号の入力を受けて、デコードされた予測ベクトルおよびデコードされた画素を動き補償回路50に供給した後に、REQ信号を1から0に変更する。そして、次に入力されたマクロブロックのデコードが終了した時点で、REQ信号を、再び0から1に変更する。

【0106】また、スライスデコーダ48のマクロブロック検出回路85乃至逆DCT回路88およびスライスデコーダ49のマクロブロック検出回路89乃至逆DCT回路92においても、スライスデコーダ47のマクロブロック検出回路81乃至逆DCT回路84と同様の処理が行われるので、その説明は省略する。

【0107】動き補償回路50は、スライスデコーダ47乃至49から入力されたデータの動き補償が終了したか否かを示すReg_REQ_A、Reg_REQ_BおよびReg_REQ_Cの3つのレジスタを有し、これらのレジスタの値を参照しながら、適宜、スライスデコーダ47乃至49のうちの1つを選んで、動き補償実行要求を受け付け（すなわち、REQ信号に対して、ACK信号を出力して、予測ベクトルと画素の入力を受ける）、動き補償処理を実行する。このとき、動き補償回路50は、スライスデコーダ47乃至49のうち、所定のタイミングにおいてREQ信号が1であるスライスデコーダ47乃至49に対する動き補償が、それぞれ1回ずつ終了した後に、次の動き補償要求を受け付ける。例えば、スライスデコーダ47が連続して動き補償要求を出しても、スライスデコーダ48およびスライスデコーダ49の動き補償が終了す

るまで、スライスデコーダ47の2つ目の動き補償要求は受け付けられない。動き補償回路50が、スライスデコーダ47乃至49のいずれのデコーダの出力に対して動き補償を実行するかを選択する処理については、図18および図19を用いて後述する。

【0108】スライスデコーダ47乃至49のいずれかから入力されるマクロブロックが動き補償を使用していない場合、動き補償回路50は、その画素データが輝度データであれば、輝度バッファ制御回路51を介して、ビデオバッファ33の輝度バッファ71に書き込み、その画素データが色差データであれば、色差バッファ制御回路52を介して、ビデオバッファ33の色差バッファ72に書き込み、表示出力に備えるとともに、この画素データが、他の画像の参照データとされる場合に備える。

【0109】また、スライスデコーダ47乃至49のいずれかから出力されるマクロブロックが動き補償を使用している場合、動き補償回路50は、スライスデコーダ47乃至49のうち対応するデコーダから入力される予測ベクトルに従って、その画素データが輝度データであれば、輝度バッファ制御回路51を介して、輝度バッファ71から参照画素を読み込み、その画素データが色差データであれば、色差バッファ制御回路52を介して、色差バッファ72から参照画素データを読み込む。そして、動き補償回路50は、読み込んだ参照画素データを、スライスデコーダ47乃至49のいずれかから供給された画素データに加算し、動き補償を行う。

【0110】動き補償回路50は、動き補償を行った画素データを、その画素データが輝度データであれば、輝度バッファ制御回路51を介して、輝度バッファ71に書き込み、その画素データが色差データであれば、色差バッファ制御回路52を介して、色差バッファ72に書き込み、表示出力に備えるとともに、この画素データが、他の画素の参照データとされる場合に備える。

【0111】表示出力回路53は、デコードした画像データを出力するための同期タイミング信号を発生し、このタイミングに従って、輝度バッファ制御回路51を介して、輝度バッファ71から輝度データを読み出し、色差バッファ制御回路52を介して、色差バッファ72から色差データを読み出して、復号ビデオ信号として出力する。

【0112】ドライブ35は、コントローラ34に接続されており、必要に応じて装着される磁気ディスク101、光ディスク102、光磁気ディスク103、および半導体メモリ104などとデータの授受を行う。また、コントローラ34は、以上説明したIC31、およびドライブ35の動作を制御するものである。コントローラ34は、例えば、ドライブに装着されている磁気ディスク101、光ディスク102、光磁気ディスク103、および半導体メモリ104などに記録されているプログ

ラムに従って、IC31に処理を実行させることができる。

【0113】次に、図16のフローチャートを参照して、スライスデコーダ制御回路46の処理について説明する。

【0114】ステップS1において、スライスデコーダ制御回路46は、処理するスライスが、符号化ストリームの何番目のスライスであるかを表わすレジスタの値をN=1とする。ステップS2において、スライスデコーダ制御回路46は、スライスデコーダ47が処理中であるか否かを判断する。

【0115】ステップS2において、スライスデコーダ47が処理中ではないと判断された場合、ステップS3において、スライスデコーダ制御回路46は、ピクチャ層のパラメータと、スタートコード情報に含まれるスライスNの書き込みポイントをスライスデコーダ47に供給し、スライスデコーダ47にスライスNをデコードさせ、処理はステップS8に進む。

【0116】ステップS2において、スライスデコーダ47が処理中であると判断された場合、ステップS4において、スライスデコーダ制御回路46は、スライスデコーダ48が処理中であるか否かを判断する。ステップS4において、スライスデコーダ48が処理中ではないと判断された場合、ステップS5において、スライスデコーダ制御回路46は、ピクチャ層のパラメータと、スタートコード情報に含まれるスライスNの書き込みポイントをスライスデコーダ48に供給し、スライスデコーダ48にスライスNをデコードさせ、処理はステップS8に進む。

【0117】ステップS4において、スライスデコーダ48が処理中であると判断された場合、ステップS6において、スライスデコーダ制御回路46は、スライスデコーダ49が処理中であるか否かを判断する。ステップS6において、スライスデコーダ49が処理中であると判断された場合、処理は、ステップS2に戻り、それ以降の処理が繰り返される。

【0118】ステップS6において、スライスデコーダ49が処理中ではないと判断された場合、ステップS7において、スライスデコーダ制御回路46は、ピクチャ層のパラメータと、スタートコード情報に含まれるスライスNの書き込みポイントをスライスデコーダ49に供給し、スライスデコーダ49にスライスNをデコードさせ、処理はステップS8に進む。

【0119】ステップS8において、スライスデコーダ制御回路46は、処理するスライスが符号化ストリームの何番目のスライスであるかを示すレジスタの値をN=N+1とする。ステップS9において、スライスデコーダ制御回路46は、全スライスのデコードが終了したか否かを判断する。ステップS9において、全スライスのデコードが終了されていないと判断された場合、処理

は、ステップS2に戻り、それ以降の処理が繰り返される。ステップS9において、全スライスのデコードが終了されたと判断された場合、処理が終了される。

【0120】図17は、図16を用いて説明したスライスデコーダ制御回路46の処理の具体例を示す図である。上述したように、ピクチャデコーダ45でピクチャ層のデータがデコードされ、そのパラメータがスライスデコーダ制御回路46に供給される。ここで、図6を用いて説明したステップS1において、スライスデコーダ制御回路46は、レジスタの値を $N=1$ とする。ステップS2において、スライスデコーダ47は処理中ではないと判断されるので、ステップS3において、スライスデコーダ制御回路46は、ピクチャ層のパラメータと、スタートコード情報に含まれるスライス1の書き込みポインタをスライスデコーダ47に供給し、スライスデコーダ47にスライス N ($N=1$) をデコードさせ、ステップS8において、レジスタの値を $N=N+1$ とする。そして、ステップS9において、全スライスのデコードが終了していないと判断されるため、処理はステップS2に戻る。

【0121】ステップS2において、スライスデコーダ47は処理中であると判断される。そして、ステップS4において、スライスデコーダ48は処理中ではないと判断されるので、ステップS5において、スライスデコーダ制御回路46は、ピクチャ層のパラメータと、スライス2の書き込みポインタを、スライスデコーダ48に供給し、スライスデコーダ48にスライス N ($N=2$) をデコードさせ、ステップS8において、 $N=N+1$ とする。そして、ステップS9において、全スライスのデコードが終了していないと判断されるため、処理はステップS2に戻る。

【0122】ステップS2において、スライスデコーダ47は処理中であると判断され、ステップS4において、スライスデコーダ48は処理中であると判断される。そして、ステップS6において、スライスデコーダ49は処理中ではないと判断されるので、ステップS7において、スライスデコーダ処理回路は、ピクチャ層のパラメータと、スライス3の書き込みポインタを、スライスデコーダ49に供給し、スライスデコーダ49にスライス N ($N=3$) をデコードさせ、ステップS8において、 $N=N+1$ とする。そして、ステップS9において、全スライスのデコードが終了していないと判断されるため、処理はステップS2に戻る。

【0123】スライスデコーダ47乃至49は、入力されたスライスのデコード処理を実施した後、デコード処理の完了を示す信号をスライスデコーダ制御回路46に出力する。すなわち、スライスデコーダ47乃至49のいずれかからスライス2のデコードの完了を示す信号が入力されるまで、スライスデコーダ47乃至49は全て処理中であるので、ステップS2、ステップS4、お

びステップS6の処理が繰り返される。そして、図17の図中Aで示されるタイミングで、スライスデコーダ48がデコード処理の完了を示す信号を、スライスデコーダ46に出力した場合、ステップS4において、スライスデコーダ48が処理中ではないと判断されるので、ステップS5において、スライスデコーダ制御回路46は、スライス4の書き込みポインタを、スライスデコーダ48に供給し、スライスデコーダ48に、スライス N ($N=4$) をデコードさせ、ステップS8において、 $N=N+1$ とする。そして、ステップS9において、全スライスのデコードが終了していないと判断されるため、処理はステップS2に戻る。

【0124】そして、次にスライスデコーダ47乃至49のいずれかからデコード処理の完了を示す信号の入力を受けるまで、スライスデコーダ制御回路46は、ステップS2、ステップS4、およびステップS6の処理を繰り返す。図17においては、スライスデコーダ制御回路46は、図中Bで示されるタイミングで、スライスデコーダ49からスライス3のデコードの終了を示す信号の入力を受けるので、ステップS6において、スライスデコーダ49は処理中ではないと判断される。ステップS7において、スライスデコーダ制御回路46は、スライス5の書き込みポインタをスライスデコーダ49に供給し、スライスデコーダ49に、スライス N ($N=5$) をデコードさせ、ステップS8において、 $N=N+1$ とする。そして、ステップS9において、全スライスのデコードが終了していないと判断されるため、処理はステップS2に戻る。以下、最後のスライスのデコードが終了されるまで、同様の処理が繰り返される。

【0125】このように、スライスデコーダ制御回路46は、スライスデコーダ47乃至49の処理状況を参照しながら、スライスのデコード処理を割り当てるので、複数のデコーダを効率よく使用することができる。

【0126】次に、図18のフローチャートを参照して、動き補償回路50による、スライスデコーダの調停処理について説明する。

【0127】ステップS21において、動き補償回路50は、内部のレジスタReg_REQ_A、Reg_REQ_BおよびReg_REQ_Cを初期化する。すなわち、Reg_REQ_A=0、Reg_REQ_B=0、Reg_REQ_C=0とする。

【0128】ステップS22において、動き補償回路50は、レジスタの値が全て0であるか否かを判断する。ステップS22において、レジスタの値が全て0ではない(すなわち、1つでも1がある)と判断された場合、処理は、ステップS24に進む。

【0129】ステップS22において、レジスタの値が全て0であると判断された場合、ステップS23において、動き補償回路50は、スライスデコーダ47乃至49から入力されるREQ信号を基に、レジスタの値を更新する。すなわち、スライスデコーダ47からREQ信

号が出力されている場合、Reg_REQ_A=1とし、スライスデコーダ48からREQ信号が出力されている場合、Reg_REQ_B=1とし、スライスデコーダ49からREQ信号が出力されている場合、Reg_REQ_C=1とする。そして処理は、ステップS24に進む。

【0130】ステップS24において、動き補償回路50は、Reg_REQ_A=1であるか否かを判断する。ステップS24において、Reg_REQ_A=1であると判断された場合、ステップS25において、動き補償回路50は、スライスデコーダ47にACK信号を送信し、Reg_REQ_A=0とする。スライスデコーダ47は、動き補償回路50に、ベクトル復号回路82で復号された予測ベクトルと、逆DCT回路84で逆DCTされた画素を出力する。そして処理は、ステップS30に進む。

【0131】ステップS24において、Reg_REQ_A=1ではないと判断された場合、ステップS26において、動き補償回路50は、Reg_REQ_B=1であるか否かを判断する。ステップS26において、Reg_REQ_B=1であると判断された場合、ステップS27において、動き補償回路50は、スライスデコーダ48にACK信号を送信し、Reg_REQ_B=0とする。スライスデコーダ48は、動き補償回路50に、ベクトル復号回路86で復号された予測ベクトルと、逆DCT回路88で逆DCTされた画素を出力する。そして処理は、ステップS30に進む。

【0132】ステップS26において、Reg_REQ_B=1ではないと判断された場合、ステップS28において、動き補償回路50は、Reg_REQ_C=1であるか否かを判断する。ステップS28において、Reg_REQ_C=1ではないと判断された場合、処理は、ステップS22に戻り、それ以降の処理が繰り返される。

【0133】ステップS28において、Reg_REQ_C=1であると判断された場合、ステップS29において、動き補償回路50は、スライスデコーダ49にACK信号を送信し、Reg_REQ_C=0とする。スライスデコーダ49は、動き補償回路50に、ベクトル復号回路90で復号された予測ベクトルと、逆DCT回路92で逆DCTされた画素を出力する。そして処理は、ステップS30に進む。

【0134】ステップS30において、動き補償回路50は、スライスデコーダ47乃至49のいずれかから入力されたマクロブロックは、動き補償を使用しているか否かを判断する。

【0135】ステップS30において、マクロブロックが動き補償を使用していると判断された場合、ステップS31において、動き補償回路50は、入力されたマクロブロックに動き補償処理を行う。すなわち、動き補償回路50は、スライスデコーダ47乃至49のうち対応するデコーダから出力される予測ベクトルに従って、その画素データが輝度データであれば、輝度バッファ制御

回路51を介して、輝度バッファ71から参照画素を読み出し、その画素データが色差データであれば、色差バッファ制御回路52を介して、色差バッファ72から参照画素データを読み出す。そして、動き補償回路50は、読み出した参照画素データを、スライスデコーダ47乃至49のいずれかから供給された画素データに加算し、動き補償を行う。

【0136】動き補償回路50は、動き補償を行った画素データを、その画素データが輝度データであれば、輝度バッファ制御回路51を介して、輝度バッファ71に書き込み、その画素データが色差データであれば、色差バッファ制御回路52を介して、色差バッファ72に書き込み、表示出力に備えるとともに、この画素データが、他の画素の参照データとされる場合に備える。そして、処理は、ステップS22に戻り、それ以降の処理が繰り返される。

【0137】ステップS30において、マクロブロックが動き補償を使用していないと判断された場合、ステップS32において、動き補償回路50は、その画素データが輝度データであれば、輝度バッファ制御回路51を介して輝度バッファ71に書き込み、その画素データが色差データであれば、色差バッファ制御回路52を介して色差バッファ72に書き込み、表示出力に備えるとともに、この画素データが、他の画像の参照データとされる場合に備える。そして、処理は、ステップS22に戻り、それ以降の処理が繰り返される。

【0138】図19は、図18を用いて説明した動き補償回路50によるデコーダの調停処理の具体例を示す図である。

【0139】図19に示すタイミングCにおいて、図18のステップS22の処理により、動き補償回路50のレジスタが全て0であると判断された場合、スライスデコーダ47乃至49は、全て、REQ信号を出力しているため、ステップS23の処理により、それぞれのレジスタの値は、Reg_REQ_A=1、Reg_REQ_B=1、Reg_REQ_C=1に更新される。そして、ステップS24の処理により、Reg_REQ_A=1であると判断されるため、ステップS25において、動き補償回路50は、スライスデコーダ47にACK信号を出力して、Reg_REQ_A=0とし、スライスデコーダ47から予測ベクトルと画素の入力を受け、動き補償1を行う。

【0140】動き補償1が終了した後、すなわち、図19のDで示されるタイミングにおいて、処理は、再びステップS22に戻る。図中Dで示されるタイミングにおいては、スライスデコーダ47から、REQ信号が出力されている。しかし、レジスタの値は、Reg_REQ_A=0、Reg_REQ_B=1、Reg_REQ_C=1であり、ステップS22において、レジスタの値は、全て0ではないと判断されるため、処理は、ステップS24に進み、レジスタの値は更新されない。

【0141】ステップS24において、Reg_REQ_A=0であると判断され、ステップS26において、Reg_REQ_B=1であると判断されるので、動き補償回路50は、ステップS27において、スライスデコーダ48にACK信号を出力して、Reg_REQ_B=0とし、スライスデコーダ48から予測ベクトルと画素の入力を受け、動き補償2を行う。

【0142】動き補償2が終了した後、すなわち、図19のEで示されるタイミングにおいて、処理は、再びステップS22に戻る。図中Eで示されるタイミングにおいても、スライスデコーダ47から、REQ信号が出力されている。しかし、レジスタの値は、Reg_REQ_A=0、Reg_REQ_B=0、Reg_REQ_C=1であるので、ステップS22において、レジスタの値は全て0ではないと判断されるので、図中Dで示されるタイミングのときと同様、レジスタの値は更新されない。

【0143】そして、ステップS24において、Reg_REQ_A=0であると判断され、ステップS26において、Reg_REQ_B=0であると判断され、ステップS28において、Reg_REQ_C=1であると判断されるので、動き補償回路50は、ステップS29において、スライスデコーダ49にACK信号を出力して、Reg_REQ_C=0とし、スライスデコーダ49から予測ベクトルと画素の入力を受け、動き補償3を行う。

【0144】動き補償3が終了した後、すなわち、図19のFで示されるタイミングにおいて、処理は、再びステップS22に戻る。Fで示されるタイミングにおいては、レジスタの値は、Reg_REQ_A=0、Reg_REQ_B=0、Reg_REQ_C=0であるので、ステップS23において、レジスタの値が更新され、Reg_REQ_A=1、Reg_REQ_B=1、Reg_REQ_C=0となる。

【0145】そして、ステップS24において、Reg_REQ_A=1であると判断され、同様の処理により、動き補償4が実行される。

【0146】このような処理を繰り返すことにより、動き補償回路50は、スライスデコーダ47乃至49を調停しながら、動き補償を行う。

【0147】以上説明したように、図15のMPEGビデオデコーダにおいては、スタートコードバッファ62を設けたことにより、ピクチャデコーダ45乃至スライスデコーダ49を、お互いの動作の終了を待つことなしに、ストリームバッファ61にアクセスさせることができる。また、スライスデコーダ47乃至49は、スライスデコーダ制御回路46の処理により、同時に動作させることができる。更に、動き補償回路30は、適宜、1つのスライスデコーダを選択し、それぞれ分離された輝度バッファ71および色差バッファ72にアクセスし、動き補償を行うことができる。したがって、図15のMPEGビデオデコーダにおいては、デコード処理性能およびバッファへのアクセス性能が向上され、2:4:4P@HL 50

に対するデコード処理が可能となる。

【0148】次に、図15のMPEGビデオデコーダに入力されたMPEGストリームが復号されて再生される場合の、フレームのバッファリングについて説明する。

【0149】図20は、図15のMPEGビデオデコーダを備えた再生装置の構成を示すブロック図である。なお、図15における場合と対応する部分には同一の符号を付してあり、その説明は適宜省略する。

【0150】ハードディスク112には、MPEG符号化ストリームが記録されている。サーボ回路111は、コントローラ34の制御に基づいて、ハードディスク112を駆動し、図示しないデータ読み取り部により読み出されたMPEGストリームがIC31の再生回路121に入力される。

【0151】再生回路121は、図15を用いて説明したストリーム入力回路41乃至クロック発生回路44を含む回路であり、順方向の再生時には、入力された順番にMPEGストリームを再生ストリームとしてMPEGビデオデコーダ122に出力する。そして、逆方向の再生（逆転再生）時には、ストリームバッファ61を用いて、入力されたMPEG符号化ストリームを、逆転再生に適した順番に並べ替えた後、再生ストリームとしてMPEGビデオデコーダ122に出力する。

【0152】MPEGビデオデコーダ122は、図15を用いて説明したピクチャデコーダ45乃至表示出力回路53を含む回路であり、動き補償回路50の処理により、必要に応じて、ビデオバッファ33に蓄積された復号されたフレームを参照画像として読み出し、動き補償を実行して、入力された再生ストリームの各ピクチャ（フレーム）を、上述した方法で復号し、ビデオバッファ33に蓄積するとともに、表示出力回路53の処理により、ビデオバッファ33に蓄積されたフレームを順次読み出して、図示しない表示部もしくは表示装置に出力し、表示させる。

【0153】ここでは、ハードディスク112に蓄積されたMPEG符号化ストリームを復号して出力し、表示させる場合を例にあげて説明したが、図15のMPEGビデオデコーダを備えた再生装置、もしくは録画再生装置は、図20で示す構成と異なる構成（例えば、ストリームバッファ61と同様に、符号化ストリームを保持する機能、および再生回路121と同様に、フレームを並べ替える機能をMPEGビデオデコーダ122に備えさせているような構成）であっても、基本的に同様の処理により、入力されたMPEG符号化ストリームが復号されて、出力される。

【0154】また、符号化ストリームを蓄積する蓄積媒体は、ハードディスク112以外にも、光ディスク、磁気ディスク、光磁気ディスク、半導体メモリ、磁気テープなど、様々な記録媒体を用いることが可能であることは言うまでもない。

【0155】図21および図22を用いて、MPEG予測符号化画像のピクチャ構成について説明する。

【0156】図21は、図示しないエンコーダ（符号化装置）に入力されて符号化されるMPEGビデオ信号のピクチャ構成を示す図である。

【0157】フレームI2は、イントラ符号化フレーム（Iピクチャ）であり、他の画像を参照することなく、符号化（エンコード）が行われる。このようなフレームは、復号の開始点である符号化シーケンスのアクセスポイントを提供するが、その圧縮率は、余り高くない。

【0158】フレームP5、フレームP8、フレームPb、およびフレームPeは、前方向予測符号化フレーム（Pピクチャ）であり、過去のIピクチャまたはPピクチャからの動き補償予測により、Iピクチャより効率的な符号化が行われる。Pピクチャ自身も、予測の参照として使用される。フレームB3、フレームB4・・・フレームBdは、双方向予測符号化フレームであり、Iピクチャ、Pピクチャと比較して、より効率的に圧縮が行われるが、動き補償のためには、過去および将来の双方の参照画像を必要とする。Bピクチャが、予測の参照として使用されることはない。

【0159】図22に、図21を用いて説明したMPEG符号化画像を生成するために、図示しないエンコーダで実行される、フレーム間予測を用いたMPEGビデオ信号の符号化の例を示す。

【0160】入力されたビデオ信号は、例えば15フレーム毎にGOP（Group of Pictures）に分割され、各GOPのはじめから3番目がIピクチャとされ、以下、3フレーム毎に出現するフレームがPピクチャとされ、それ以外のフレームがBピクチャとされる（M=15、N=3）。そして、符号化に後方予測を必要とするBピクチャであるフレームB10およびフレームB11がバッファに一時保存され、IピクチャであるフレームI12が最初に符号化される。

【0161】フレームI12の符号化の終了後、バッファに一時保存されたフレームB10およびフレームB11が、フレームI12を参照画像として符号化される。Bピクチャは、本来、過去と将来の両方の参照画像を参照して符号化されるべきであるが、フレームB10およびフレームB11のように、前方向に参照可能な画像がない場合、Closed GOPフラグを立てて、前方向予測をせずに後方向の予測のみで符号化が行われる。

【0162】フレームB10およびフレームB11の符号化が行われている間に入力されたフレームB13およびフレームB14は、ビデオバッファに蓄えられ、次に入力されたフレームP15が、フレームI12を前方向予測画像として参照して、符号化される。そして、ビデオバッファから読み出されたフレームB13およびフレームB14は、フレームI12を前方向予測画像として参照し、フレームP15を後方向予測画像として参照し

て符号化される。

【0163】続いて、フレームB16およびフレームB17がビデオバッファに蓄えられ、以下、順次同様にして、Pピクチャは、前回符号化されたIピクチャ、もしくはPピクチャを前方向予測画像として参照して符号化され、Bピクチャは、一旦ビデオバッファに蓄えられた後、以前に符号化されたIピクチャ、もしくはPピクチャを、前方向予測画像もしくは後方向予測画像として参照して符号化される。

10 【0164】このようにして、複数のGOPに渡って、画像データが符号化され、符号化ストリームが生成される。図20のハードディスク112には、上述した方法で符号化されたMPEG符号化ストリームが記録される。

【0165】符号化時に、DCT変換により得られたDCT係数行列は、通常の画像をDCT変換した場合、低周波成分において大きく、高周波成分では小さいという特徴がある。この特徴を利用し情報の圧縮を図るのが量子化（各DCT係数に対してある量子化単位で割り、小数点以下を丸める）である。量子化単位は8×8の量子化テーブルとして設定され、低周波成分に対しては小さい値、高周波成分に対しては大きい値が設定される。量子化の結果、行列の左上以外の成分はほとんど0になる。そして、量子化マトリクスに対応する量子化IDが、圧縮データに付加されて復号側に渡される。すなわち、図20のMPEGビデオデコーダ122は、量子化IDから量子化マトリクスを参照して、MPEG符号化ストリームを復号する。

【0166】次に、図23を用いて、ハードディスク112から、順方向にビデオデータを再生する場合に、GOP1乃至GOP3の符号化ストリームが再生回路121に入力され、MPEGビデオデコーダ122の処理により復号される処理について説明する。

【0167】順方向再生のためにハードディスク112から再生回路121に入力されるMPEGビデオストリームは、再生回路121の処理により、入力された順番と同一のピクチャ配列の再生ストリームとして、MPEGビデオデコーダ122に出力される。MPEGビデオデコーダ122において、再生ストリームは、図15乃至図19を用いて説明した手順に従って復号され、ビデオバッファ33に蓄積される。

【0168】最初に入力されたフレームI12は、Iピクチャであるので、復号に参照画像を必要としない。MPEGビデオデコーダ122において復号されたフレームI12が蓄積されるビデオバッファ33内のバッファ領域をバッファ1とする。

【0169】次にMPEGビデオデコーダ122に入力されるフレームB10およびフレームB11は、Bピクチャであるが、Closed GOPフラグが立っているので、ビデオバッファ33のバッファ1に蓄積されているフレームI12を後方向参照画像として参照して復号され、ビデオ

バッファ33に蓄積される。復号されたフレームB10が蓄積されるバッファ領域をバッファ3とする。

【0170】そして、表示出力回路53の処理により、ビデオバッファ33のバッファ3からフレームB10が読み出されて、図示しない表示部に出力されて表示される。そして、次に復号されたフレームB11が、ビデオバッファ33のバッファ3に蓄積（すなわち、バッファ3に上書き）されたのち、読み出されて、図示しない表示部に出力されて、表示される。

【0171】その次に、バッファ1からフレームI12が読み出されて、図示しない表示部に出力されて、表示され、そのタイミングで、次のフレームP15が、ビデオバッファ33のバッファ1に蓄積されているフレームI12を参照画像として復号され、ビデオバッファ33のバッファ2に蓄積される。

【0172】フレームB10およびフレームB11にClosed GOPフラグが立っていない場合、前方向に参照できる画像がないため、フレームB10およびフレームB11は、復号されない。そのような場合、フレームI12が一番初めに表示出力回路53から出力され、表示される。

【0173】次に入力されるフレームB13は、ビデオバッファ33のバッファ1に蓄積されているフレームI12を前方向参照画像とし、バッファ2に蓄積されているフレームP15を後方向参照画像として参照して復号され、バッファ3に蓄積される。そして、表示出力回路53の処理により、ビデオバッファ33のバッファ3からフレームB13が読み出されて、出力表示処理が実行されている間に、次に入力されるフレームB14が、ビデオバッファ33のバッファ1に蓄積されているフレームI12を前方向参照画像とし、バッファ2に蓄積されているフレームP15を後方向参照画像として参照して復号され、バッファ3に蓄積される。そして、表示出力回路53の処理により、ビデオバッファ33のバッファ3からフレームB14が読み出されて、出力され、表示される。

【0174】次に入力されるフレームP18は、バッファ2に蓄積されているフレームP15を前方向参照画像として復号される。フレームB14の復号が終われば、バッファ1に蓄積されているフレームI12は、その後、参照されることがないので、復号されたフレームP18は、ビデオバッファ33のバッファ1に蓄積される。そして、バッファ1にフレームP18が蓄積されるタイミングで、バッファ2からフレームP15が読み出されて、出力され、表示される。

【0175】以下、同様にして、GOP1のフレームが、順次復号され、バッファ1乃至3に蓄積され、順次、読み出されて表示される。

【0176】GOP2の先頭のフレームI22が入力されたとき、IピクチャであるフレームI22は、復号時

に参照画像を必要としないので、そのまま復号され、バッファ2に蓄積される。そのタイミングで、GOP1のフレームP1eが読み出されて、出力され、表示される。

【0177】続いて入力されるGOP2のフレームB20およびフレームB21は、バッファ1のフレームP1eを前方向参照画像、バッファ2のフレームI22を後方向参照画像として復号され、バッファ3に順次蓄積され、読み出されて、表示される。このように、GOPの先頭にあるBピクチャは、前のGOPのPピクチャを前方向参照画像として復号される。

【0178】以下、同様にして、GOP2のフレームが、順次復号され、バッファ1乃至3に蓄積され、順次、読み出されて表示される。そして、同様にして、GOP3以下のそれぞれのフレームが順次復号され、バッファ1乃至3に蓄積され、順次、読み出されて表示される。

【0179】以上の処理において、MPEGビデオデコーダ122は、量子化IDを参照して復号処理を実行する。

【0180】次に、図20を用いて説明した再生装置において、逆転再生を実施する場合について説明する。

【0181】従来の逆転再生においては、Iピクチャのみを取り出して復号を実行していたので、15フレーム中の1フレームしか表示されない、不自然な再生画像しか得ることができなかった。

【0182】それに対して、図20の再生回路121には、スタートコードバッファ62に記録されたスタートコードに基づいて、ストリームバッファ61に入力されたGOPのフレームの順番を変更して再生ストリームを生成させることができ、MPEGビデオデコーダ122には、15フレーム全てを復号させることができる。

【0183】しかしながら、再生回路121は、逆転再生を行うために、スタートコードバッファ62に記録されたスタートコードに基づいて、ストリームバッファ61に入力されたGOPのフレームの順番を単純に逆転させて再生ストリームを生成すればいいのではない。

【0184】例えば、図22を用いて説明したMPEG符号化ストリームのGOP2およびGOP1を逆転再生させる場合、最初に表示されて表示されるフレームは、フレームP2eでなければならないが、フレームP2eを復号するためには、前方向参照画像として、フレームP2bを参照する必要があり、更にフレームP2bを復号するためには、前方向参照画像としてフレームP28が必要である。フレームP28を復号するためにも前方向参照画像が必要であるので、結局、フレームP2eを復号し、出力して表示させるためには、GOP2のIピクチャおよびPピクチャが全て復号されていなければならない。

【0185】逆転再生時に最初に表示されるフレームP2eを復号するために、GOP2を全て復号させてビデオバッファ33に蓄積させ、後のフレームから順に読み出

す方法も考えられるが、その場合、ビデオバッファ33には、1 GOP (15フレーム) 分のバッファ領域が必要となる。

【0186】また、この方法では、フレームP2eから、フレームI22までは復号して再生することが可能であるが、GOP2のはじめの2フレーム、すなわち、逆転再生時に最後に表示されるべきフレームであるフレームB21およびフレームB20を復号するためには、前方向参照画像としてGOP1のフレームP1eが必要である。GOP1のフレームP1eを復号するためには、GOP1 10の全てのIピクチャおよびPピクチャが必要である。

【0187】すなわち、この方法では、ビデオバッファ33に15フレーム分のバッファ領域を必要としながら、1 GOP分の全てのフレームの逆転再生ができない。

【0188】図22を用いて説明したように、M=15、N=3で符号化を行った場合、1 GOP内には、合計5フレームのIピクチャもしくはPピクチャが含まれる。

【0189】そこで、ストリームバッファ61に、少なくとも2 GOP分のフレームを蓄積することが可能なようにし、再生回路121において生成される再生ストリームのフレームの順番を、MPEGビデオデコーダ122の逆転再生のためのデコードの順番に基づいて決定し、ビデオバッファ33に、少なくとも、「1 GOPに含まれるIピクチャおよびPピクチャの合計数+2」で表わされる数のフレーム (例えば、M=15、N=3のMPEG符号化ストリームを逆転再生する場合、7フレーム) を蓄積することが可能なようにすることにより、GOPをまたいだ部分も連続して、全てのフレームを逆転再生することができる。

【0190】図24を用いて、ハードディスク112から、GOP1乃至GOP3の画像データが逆転再生される場合の復号処理について説明する。

【0191】コントローラ34は、サーボ回路111を制御して、ハードディスク112から、まずGOP3、次いでGOP2のMPEG符号化ストリームを再生回路121に出力させる。再生回路121は、ストリームバッファ61に、GOP3、次いでGOP2のMPEG符号化ストリームを蓄積させる。

【0192】再生回路121は、ストリームバッファ61からGOP3の先頭フレーム132を読み出し、再生ストリームの最初のフレームとして、MPEGビデオデコーダ122に出力する。フレーム132はIピクチャであるから、復号のために参照画像を必要としないので、MPEGビデオデコーダ122において復号され、ビデオバッファ33に蓄積される。ビデオバッファ33において、復号されたフレーム132が蓄積される領域を、バッファ1とする。

【0193】ここで、それぞれのフレームのデータは、図2を用いて説明したヘッダーおよび拡張データに記載 50

されたパラメータを基に復号される。上述したように、MPEGビデオデコーダ122のピクチャデコーダ45において、それぞれのパラメータがデコードされ、スライスデコーダ制御回路46に供給されて、復号処理に用いられる。GOP1が復号される場合、GOP1のsequence_header、sequence_extension、およびGOP_headerに記載されている上位層のパラメータ (例えば、上述した量子化マトリクス) を用いて復号が実施され、GOP2が復号される場合、GOP2のsequence_header、sequence_extension、およびGOP_headerに記載されている上位層のパラメータを用いて復号が実施され、GOP3が復号される場合、GOP3のsequence_header、sequence_extension、およびGOP_headerに記載されている上位層のパラメータを用いて復号が実施される。

【0194】しかしながら、逆転再生時には、GOP毎に復号が実施されないで、MPEGビデオデコーダ122は、それぞれのGOPにおいて、最初にIピクチャがデコードされた時に、上位層パラメータをコントローラ34に供給する。コントローラ34は、内部に有する図示しないメモリに、供給された上位層パラメータを保持する。

【0195】コントローラ34は、MPEGビデオデコーダ122において実行される復号処理を監視し、復号処理中のフレームに対応する上位層パラメータを内部のメモリから読み出してMPEGビデオデコーダ122に供給し、適切な復号処理がなされるようにする。

【0196】図24中、再生ストリームのフレーム番号の上部に示される数字は、量子化IDであり、再生ストリームの各フレームは、図23を用いて説明した順方向の復号と同様に、量子化IDを基に復号される。

【0197】なお、本実施の形態においては、コントローラ34がその内部にメモリを有し、上位層符号化パラメータを保持するものとして説明しているが、コントローラ34に接続されたメモリを設けるようにし、コントローラ34が内部にメモリを保有せず、外部のメモリに上位層符号化パラメータを保持し、必要に応じて読み出して、MPEGビデオデコーダ122に供給することができるとしても良い。

【0198】また、MPEGビデオデコーダ122に、GOPの上位層符号化パラメータを保持するためのメモリを備えさせるようにしてもよい。更に、上位層符号化パラメータなどの符号化条件が既知である場合、MPEGビデオデコーダ122に、符号化条件を予め設定しても良いし、上位層符号化パラメータがGOPによって変化しないことが既知である場合、コントローラ34によって、GOP毎に上位層符号化パラメータを読み出して、フレーム毎にMPEGビデオデコーダ122に設定するのではなく、動作開始時に一度だけ、MPEGビデオデコーダ122に符号化パラメータを設定するようにしても良い。

【0199】再生回路121は、ストリームバッファ6

1からフレームP35を読み出し、再生ストリームの次のフレームとして、MPEGビデオデコーダ122に出力する。MPEGビデオデコーダ122において、フレームP35は、バッファ1に記録されているフレームI32を前方向参照画像として復号され、ビデオバッファ33に蓄積される。ビデオバッファ33において、復号されたフレームP35が蓄積される領域を、バッファ2とする。

【0200】再生回路121は、ストリームバッファ61からフレームP38、フレームP3b、およびフレームP3eを順次読み出し、再生ストリームとして出力する。これらのPピクチャは、MPEGビデオデコーダ122において、一つ前に復号されたPピクチャを前方向参照画像として復号され、ビデオバッファ33に蓄積される。ビデオバッファ33において、復号されたこれらのPピクチャのフレームが蓄積される領域を、バッファ3乃至バッファ5とする。

【0201】このとき、GOP3のIピクチャおよびPピクチャは、全て復号され、ビデオバッファ33に蓄積されている状態である。

【0202】続いて、再生回路121は、ストリームバッファ61からGOP2のフレームI22を読み出し、再生ストリームとして出力する。MPEGデコーダ122において、IピクチャであるフレームI22は、参照画像を必要とせずに復号され、ビデオバッファ33に蓄積される。復号されたフレームI22が蓄積される領域をバッファ6とする。また、バッファ6にフレームI22が蓄積されるタイミングで、バッファ5から、GOP3のフレームP3eが読み出されて、出力され、逆転再生の最初の画像として表示される。

【0203】再生回路121は、ストリームバッファ61から、GOP3のフレームB3d、すなわち、GOP3のBピクチャの中で、初めに逆転再生されるべきフレームを読み出し、再生ストリームとして出力する。MPEGデコーダ122において、フレームB3dは、バッファ4のフレームP3bを前方向参照画像、バッファ5のフレームP3eを後方向参照画像として復号され、ビデオバッファ33に蓄積される。復号されたフレームB3dが蓄積される領域をバッファ7とする。

【0204】バッファ7に蓄積されたフレームB3dは、フレーム/フィールド変換および出力ビデオ同期タイミングへのタイミング合わせが行われたあと、出力され、表示される。フレームB3dの表示と同一のタイミングで、再生回路121は、ストリームバッファ61から、GOP3のフレームB3cを読み出して、MPEGビデオデコーダ122に出力する。MPEGビデオデコーダ122において、フレームB3cは、フレームB3dと同様に、バッファ4のフレームP3bを前方向参照画像、バッファ5のフレームP3eを後方向参照画像として復号される。

【0205】先に復号され、出力されたフレームB3d

は、Bピクチャであるので、他のフレームの復号のために参照されることはない。したがって、復号されたフレームP3cは、フレームB3dに代わって、バッファ7に蓄積（すなわち、バッファ7に上書き）され、フレーム/フィールド変換および出力ビデオ同期タイミングへのタイミング合わせが行われたあと、出力され、表示される。

【0206】再生回路121は、ストリームバッファ61から、GOP2のフレームP25を読み出して、MPEGビデオデコーダ122に出力する。MPEGビデオデコーダ122において、GOP2のフレームP25は、バッファ6のフレームI22を前方向参照画像として復号される。バッファ5に蓄積されているフレームP3eは、これ以降、参照画像として利用されることはないので、復号されたフレームP25は、フレームP3eに代わって、バッファ5に蓄積される。そして、バッファ5にフレームP25が蓄積されるのと同じタイミングで、バッファ4のフレームP3bが読み出され、表示される。

【0207】再生回路121は、ストリームバッファ61から、GOP3のフレームB3aを読み出し、再生ストリームとして出力する。MPEGデコーダ122において、フレームB3aは、バッファ3のフレームP38を前方向参照画像、バッファ4のフレームP3bを後方向参照画像として復号され、ビデオバッファ33のバッファ7に蓄積される。

【0208】バッファ7に蓄積されたフレームB3aは、フレーム/フィールド変換および出力ビデオ同期タイミングへのタイミング合わせが行われたあと、出力され、表示される。フレームB3aの表示と同一のタイミングで、再生回路121は、ストリームバッファ61から、GOP3のフレームB39を読み出して、MPEGビデオデコーダ122に出力する。MPEGビデオデコーダ122において、フレームB39は、フレームB3aと同様に、バッファ3のフレームP38を前方向参照画像、バッファ4のフレームP3bを後方向参照画像として復号され、フレームB3aに代わって、バッファ7に蓄積され、フレーム/フィールド変換および出力ビデオ同期タイミングへのタイミング合わせが行われたあと、出力され、表示される。

【0209】再生回路121は、ストリームバッファ61から、GOP2のフレームP28を読み出して、MPEGビデオデコーダ122に出力する。MPEGビデオデコーダ122において、GOP2のフレームP28は、バッファ5のフレームP25を前方向参照画像として復号される。バッファ4に蓄積されているフレームP3bは、これ以降、参照画像として利用されることはないので、復号されたフレームP28は、フレームP3bに代わって、バッファ4に蓄積される。そして、バッファ4にフレームP28が蓄積されるのと同じタイミングで、バッファ3のフレームP38が読み出され、表示される。

【0210】このように、GOP2のIピクチャもしくはPピクチャが復号され、バッファ33に蓄積されるタイミングで、GOP3のIピクチャもしくはPピクチャが、バッファ33から読み出されて、表示される。

【0211】以下、同様にして、図24に示されるように、GOP3の残りのBピクチャ、およびGOP2の残りのPピクチャが、B37、B36、P2b、B34、B33、P2eの順に復号される。復号されたBピクチャは、バッファ7に蓄積され、順次読み出されて表示される。復号されたGOP2のPピクチャは、順次、参照が終了したフレームが蓄積されていたバッファ1乃至6のいずれかに蓄積され、バッファ1乃至6のいずれかに既に蓄積されているGOP3のPピクチャが、そのタイミングで、逆転再生の順番にかなうように、Bピクチャの間に読み出されて出力される。

【0212】再生回路121は、ストリームバッファ61から、GOP3のフレームB31、次いで、フレームB30を読み出して、MPEGビデオデコーダ122に出力する。MPEGビデオデコーダ122において、フレームB31およびフレームB30の復号に必要な前方向参照画像であるフレームP2eはバッファ2に、後方向参照画像であるI32は、バッファ1にそれぞれ蓄積されているので、GOP3のはじめの2フレーム、すなわち、逆転再生時に最後に表示されるべきフレームも、復号することが可能となる。

【0213】復号されたフレームB31およびフレームB30は、順次、バッファ7に蓄積され、フレーム/フィールド変換および出力ビデオ同期タイミングへのタイミング合わせが行われたあと、出力され、表示される。

【0214】GOP3の全てのフレームがストリームバッファ61から読み出された後、コントローラ34は、サーボ回路111を制御して、ハードディスク112からGOP1を読み出させ、再生回路121に供給させる。再生回路121は、所定の処理を実行し、GOP1のスタートコードを抽出して、スタートコードバッファ62に記録させるとともに、GOP1の符号化ストリームをストリームバッファ61に供給して蓄積させる。

【0215】次に、再生回路121は、ストリームバッファ61から、GOP1のフレームI12を読み出し、再生ストリームとして、MPEGビデオデコーダ122に出力する。フレームI12は、Iピクチャであるから、MPEGビデオデコーダ122において、他の画像を参照せずに復号され、この後の処理で参照されることのないバッファ1のフレームI32に代わって、バッファ1に出力され、蓄積される。このとき、バッファ2から、フレームP2eが読み出されて、出力され、GOP2の逆転再生表示が開始される。

【0216】次に、再生回路121は、ストリームバッファ61から、GOP2のフレームB2d、すなわち、GOP2のBピクチャの中で、初めに逆転再生されるべきフレ

ームを読み出し、再生ストリームとして出力する。MPEGデコーダ122において、フレームB2dは、バッファ3のフレームP2bを前方向参照画像、バッファ2のフレームP2eを後方向参照画像として復号され、ビデオバッファ33に蓄積される。復号されたフレームB2dは、バッファ7に蓄積され、フレーム/フィールド変換および出力ビデオ同期タイミングへのタイミング合わせが行われたあと、出力され、表示される。

【0217】以下、同様にして、GOP2の、残りのBピクチャ、およびGOP1の残りのPピクチャが、B2c、P15、B2a、B29、P18、B27、B26、P1b、B24、B23、P1e、P21、P20の順に復号され、順次、参照が終了したフレームが蓄積されていたバッファ1乃至7のいずれかに蓄積され、逆転再生の順番で読み出されて出力される。そして、図示はされていないが、最後に、GOP1の残りのBピクチャが復号され、順次、バッファ7に蓄積され、逆転再生の順番で読み出されて出力される。

【0218】図24を用いて説明した処理においては、通常再生と同一速度の逆転再生を実行したが、再生回路121が、再生ストリームを通常再生時の3分の1の速度でMPEGビデオデコーダ122に出力し、MPEGビデオデコーダ122が、通常の3フレームの処理時間で1フレームのみの復号処理を実行して、図示しない表示部、もしくは表示装置に、通常の3フレームの表示時間に、同一のフレームを表示させるようにすることにより、3分の1倍速の順方向再生、および逆転再生が、同様の処理によって可能となる。

【0219】また、表示出力回路53が、同一のフレームを繰り返して出力することにより、いわゆるスチル再生も可能となる。なお、再生装置121からMPEGビデオデコーダ122へのデータ出力速度、およびMPEGビデオデコーダ122の処理速度を変更することにより、任意のnで、n分の1倍速の順方向再生、および逆転再生が、同様の処理によって可能となる。

【0220】すなわち、本発明を用いた再生装置においては、等倍速の逆転再生、n分の1倍速の逆転再生、スチル再生、n分の1倍速の順方向再生、等倍速の順方向再生の間での任意の速度において、滑らかなトリック再生が可能となる。

【0221】また、MPEGビデオデコーダ122は、MPEG2 4:2:2 P@HL対応のデコーダであるので、MPEG2 MP@MLの符号化ストリームを6倍速で復号する能力を有する。したがって、再生回路121が、MPEG2 MP@MLの符号化ストリームから生成した再生ストリームを、通常再生時の6倍の速度でMPEGビデオデコーダ122に出力すれば、MPEGビデオデコーダ122は、MPEG2 MP@MLの符号化ストリームを6倍速で復号する能力を有するので、図示しない表示部、もしくは表示装置に、6フレーム毎に抽出されたフレームを表示させるように

することにより、6倍速の順方向再生、および逆転再生が、同様の処理によって可能となる。

【0222】すなわち、本発明を用いた再生装置においては、MPEG2 MP@MLの符号化ストリームを、6倍速の逆転再生、等倍速の逆転再生、n分の1倍速の逆転再生、スチル再生、n分の1倍速の順方向再生、等倍速の順方向再生、6倍速の順方向再生の間での任意の速度において、滑らかなトリック再生が可能となる。

【0223】なお、MPEGビデオデコーダ122が、N倍速の復号能力を備えている場合、本発明の再生装置は、同様の処理により、N倍速の順方向再生、および逆転再生が可能であり、N倍速の逆転再生、等倍速の逆転再生、n分の1倍速の逆転再生、スチル再生、n分の1倍速の順方向再生、等倍速の順方向再生、N倍速の順方向再生の間での任意の速度において、滑らかなトリック再生が可能となる。

【0224】これにより、例えば、映像信号の検証時において、映像素材の内容を容易に検証することができ、映像素材検証作業の効率を改善したり、映像信号の編集作業において、編集点を快適に検索することができ、編集作業の効率を改善することができる。

【0225】上述した一連の処理は、ソフトウェアにより実行することもできる。そのソフトウェアは、そのソフトウェアを構成するプログラムが、専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、記録媒体からインストールされる。

【0226】この記録媒体は、図15もしくは図20に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク101（フロッピー（登録商標）ディスクを含む）、光ディスク102（CD-ROM（Compact Disk Read Only Memory）、DVD（Digital Versatile Disk）を含む）、光磁気ディスク103（MD（Mini Disk）を含む）、もしくは半導体メモリ104などよりなるパッケージメディアなどにより構成される。

【0227】また、本明細書において、記録媒体に記録されるプログラムを記述するステップは、記載された順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

【0228】

【発明の効果】本発明の第1の復号装置、復号方法、およびプログラムによれば、符号化ストリームを復号し、復号処理を並行して動作するようにしたので、実現可能な回路規模で実時間動作が可能な4:2:2 P@HLに対応したビデオデコーダを実現することができる。

【0229】本発明の第2の復号装置、復号方法、およびプログラムによれば、符号化ストリームを複数のス

イスデコーダで復号し、複数のスライスデコーダにより復号処理を並行して動作するようにしたので、実現可能な回路規模で実時間動作が可能な4:2:2 P@HLに対応したビデオデコーダを実現することができる。

【0230】本発明の第3の復号装置、復号方法、およびプログラムによれば、ソース符号化ストリームのピクチャを構成するスライス毎に、ソース符号化ストリームをデコードし、複数のスライスデコーダのデコードステータスを監視するとともに、複数のスライスデコーダを制御し、ピクチャに含まれるスライスの順番に関係なく、スライスデコーダにおいて実行されるデコード処理が最も早くなるように、スライスを複数のスライスデコーダに割り当てるようにしたので、実現可能な回路規模で実時間動作が可能な4:2:2 P@HLに対応したビデオデコーダを実現することができる。

【0231】本発明の第4の復号装置、復号方法、およびプログラムによれば、ソース符号化ストリームのピクチャを構成するスライス毎に、ソース符号化ストリームをデコードし、複数のスライスデコーダのデコードステータスを監視するとともに、複数のスライスデコーダを制御し、ピクチャに含まれるスライスの順番に関係なく、複数のスライスデコーダのうちデコードが終了したスライスデコーダに、デコードすべきスライスを割り当て割り当てるようにしたので、実現可能な回路規模で実時間動作が可能な4:2:2 P@HLに対応したビデオデコーダを実現することができる。

【図面の簡単な説明】

【図1】MPEG2の、プロファイルとレベルによる、各パラメータの上限値を説明するための図である。

【図2】MPEG2ビットストリームの階層構造を説明するための図である。

【図3】マクロブロック層を説明するための図である。

【図4】sequence_headerのデータ構造を説明するための図である。

【図5】sequence_extensionのデータ構造を説明するための図である。

【図6】GOP_headerのデータ構造を説明するための図である。

【図7】picture_headerのデータ構造を説明するための図である。

【図8】picture_coding_extensionのデータ構造を説明するための図である。

【図9】picture_dataのデータ構造を説明するための図である。

【図10】sliceのデータ構造を説明するための図である。

【図11】macroblockのデータ構造を説明するための図である。

【図12】macroblock_modesのデータ構造を説明するための図である。

【図13】スタートコードを説明するための図である。

【図14】従来のML@MPの符号化ストリームをデコードするビデオデコーダの構成を示すブロック図である。

【図15】本発明を適応したビデオデコーダの構成を示すブロック図である。

【図16】スライスデコーダ制御回路の処理について説明するためのフローチャートである。

【図17】スライスデコーダ制御回路の処理の具体例を説明するための図である。

【図18】動き補償回路によるスライスデコーダの調停 10 処理を説明するためのフローチャートである。

【図19】動き補償回路によるスライスデコーダの調停処理の具体例を説明するための図である。

【図20】図15のMPEGビデオデコーダを備えた再生装置の構成を示すブロック図である。

【図21】エンコーダに入力されて符号化されるMPEGビデオ信号のピクチャ構成を示す図である。

【図22】フレーム間予測を用いたMPEGの画像符号化の*

【図1】

Profile and Level	上限				
	Bit rates(Mbit/s)	Samples/line	Lines/Frame	Frames/sec	Samples/sec
4:2:2P@HL	300	1920	1152	60	62,668,800
4:2:2P@ML	50	720	608	30	11,059,200
MP@HL	80	1920	1152	60	62,668,800
MP@HL-1440	60	1440	1152	60	47,001,600
MP@ML	15	720	576	30	10,368,000
MP@LL	4	352	288	30	3,041,280
SP@ML	15	720	576	30	10,368,000

図1

【図2】

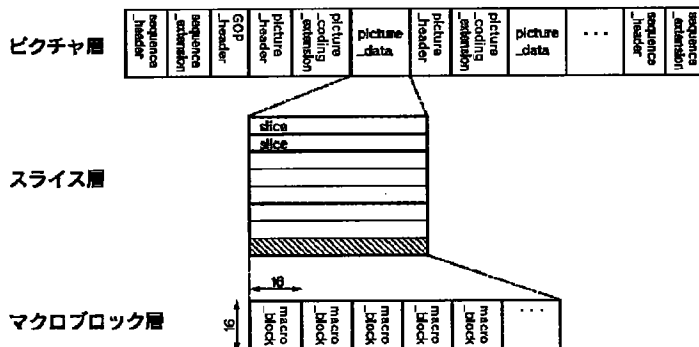


図2

* 例を示す図である。

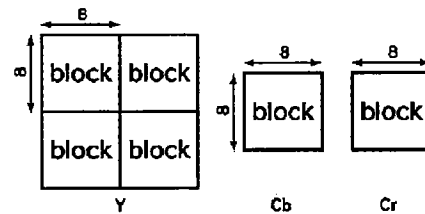
【図23】MPEG符号化ストリームが順方向に再生される場合の復号処理について説明するための図である。

【図24】MPEG符号化ストリームが逆転再生される場合の復号処理について説明するための図である。

【符号の説明】

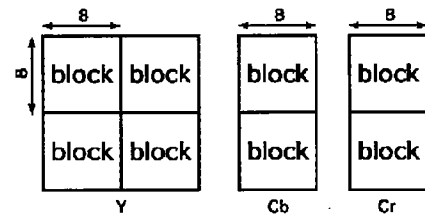
31 IC, 32 バッファ, 33 ビデオバッファ, 34 コントローラ, 42 スタートコード検出回路, 43 ストリームバッファ制御回路, 45 ピクチャデコーダ, 46 スライスデコーダ制御回路, 47乃至49 スライスデコーダ, 50 動き補償回路, 51 輝度バッファ制御回路, 52 色差バッファ制御回路, 61 ストリームバッファ, 62 スタートコードバッファ, 71 輝度バッファ, 72 色差バッファ, 111 サーボ回路, 112 ハードディスク, 121 再生回路, 122 MPEGビデオデコーダ

【図3】



4:2:0方式のmacro block層

(A)



4:2:2方式のmacro block層

(B)

図3

【図4】

sequence_header () {	ビット数	ニーモニック
sequence_header_code	32	bslbf
horizontal_size_value	12	uimsbf
vertical_size_value	12	uimsbf
aspect_ratio_information	4	uimsbf
frame_rate_code	4	uimsbf
bit_rate_value	18	uimsbf
marker_bit	1	"1"
vbv_buffer_size_value	10	uimsbf
constrained_parameters_flag	1	
load_intra_quantiser_matrix	1	
if(load_intra_quantiser_matrix)		
intra_quantiser_matrix[64]	8*64	uimsbf
load_non_intra_quantiser_matrix	1	
if(load_non_intra_quantiser_matrix)		
non_intra_quantiser_matrix[64]	8*64	uimsbf
next_start_code ()		
}		

sequence_header

図4

【図6】

group_of_picture_header () {	ビット数	ニーモニック
group_start_code	32	bslbf
time_code	25	bslbf
closed_gop	1	uimsbf
broken_link	1	uimsbf
next_start_code ()		
}		

GOP_header

図6

【図9】

picture_data () {	ビット数	ニーモニック
do {		
slice ()		
} while(nextbits()==slice_start_code)		
next_start_code()		
}		

picture_data

図9

【図5】

sequence_extension () {	ビット数	ニーモニック
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
profile_and_level_indication	8	uimsbf
progressive_sequence	1	uimsbf
chroma_format	2	uimsbf
horizontal_size_extension	2	uimsbf
vertical_size_extension	2	uimsbf
bit_rate_extension	12	uimsbf
marker_bit	1	bslbf
vbv_buffer_size_extension	8	uimsbf
low_delay	1	uimsbf
frame_rate_extension_n	2	uimsbf
frame_rate_extension_d	5	uimsbf
next_start_code ()		
}		

sequence_extension

図5

【図7】

picture_header () {	ビット数	ニーモニック
picture_start_code	32	bslbf
temporal_reference	10	uimsbf
picture_coding_type	3	uimsbf
vbv_delay	16	uimsbf
if(picture_coding_type==2 picture_coding_type==3){		
full_pel_forward_vector	1	
forward_f_code	3	uimsbf
}		
if(picture_coding_type==3)		
full_pel_forward_vector	1	
backward_f_code	3	uimsbf
}		
while(nextbits()=="I") {		
extra_bit_picture/*with the value "I"*/	1	uimsbf
extra_information_picture	8	
}		
extra_bit_picture/*with the value "0"*/	1	uimsbf
next_start_code ()		
}		

picture_header

図7

【図13】

名 称	Start code value
Picture_start_code	00
Slice_start_code	01 ~ AF
Reserved	B0
Reserved	B1
User_data_start_code	B2
Sequence_header_code	B3
Sequence_error_code	B4
Extension_start_code	B5
Reserved	B6
Sequence_end_code	B7
Group_start_code	B8
System_start_code	B9 ~ FF

図13

【図8】

picture_coding_extension()	ビット数	ニーモニック
extension_start_code	32	bslbf
extension_start_code_identifier	4	ulmsbf
f_code[0][0]/*forward horizontal*/	4	ulmsbf
f_code[0][1]/*forward vertical*/	4	ulmsbf
f_code[1][0]/*backward horizontal*/	4	ulmsbf
f_code[1][1]/*backward vertical*/	4	ulmsbf
intra_dc_precision	2	ulmsbf
picture_structure	2	ulmsbf
top_field_first	1	ulmsbf
frame_pred_frame_dct	1	ulmsbf
concealment_motion_vectors	1	ulmsbf
q_scale_type	1	ulmsbf
intra_vlc_format	1	ulmsbf
alternate_scan	1	ulmsbf
repeat_first_field	1	ulmsbf
chroma_420_type	1	ulmsbf
progressive_frame	1	ulmsbf
composite_display_flag	1	ulmsbf
if(composite_display_flag) {		
v_axis	1	ulmsbf
field_sequence	3	ulmsbf
sub_carrier	1	ulmsbf
burst_amplitude	7	ulmsbf
sub_carrier_phase	8	ulmsbf
}		
next_start_code()		
}		

picture_coding_extension

図8

【図11】

macroblock()	ビット数	ニーモニック
while (nextbits() != '0000 0001 000')		
macroblock_escape	11	bslbf
macroblock_address_increment	1-11	viclbf
macroblock_modes()		
if(macroblock_quant)		
quantiser_scale_code	5	ulmsbf
if(macroblock_motion_forward (macroblock_intra && concealment_motion_vectors))		
motion_vectors(0)		
if(macroblock_motion_backward)		
motion_vectors(1)		
if(macroblock_intra && concealment_motion_vectors)		
marker_bit	1	bslbf
if(macroblock_pattern)		
coded_block_pattern()		
for(1=0; i<block_count; i++) {		
block(i)		
}		
}		

macroblock

図11

【図10】

slice()	ビット数	ニーモニック
slice_start_code	32	bslbf
if(vertical_size>2800)		
slice_vertical_position_extension	3	ulmsbf
if(<sequence_scalable_extension() is present in the bitstream>)		
if(<scalable_mode=="data partitioning">)		
priority_breakpoint	7	ulmsbf
quantiser_scale_code	5	ulmsbf
if(nextbits() == "I") {		
intra_slice_flag	1	ulmsbf
intra_slice	1	ulmsbf
reserved_bits	7	ulmsbf
while(nextbits() == "I") {		
extra_bit_slice/*with the value "I"*/	1	ulmsbf
extra_information_slice	8	ulmsbf
}		
extra_bit_slice/*with the value "0"*/	1	ulmsbf
do {		
macroblock()		
while (nextbits() != '000 0000 0000 0000 0000')		
next_start_code()		
}		

slice

図10

【図12】

macroblock_modes()	ビット数	ニーモニック
macroblock_type	1-9	viclbf
if((spatial_temporal_weight_code_flag==1)&& (spatial_temporal_weight_code_table_index != "00")) {		
spatial_temporal_weight_code	2	ulmsbf
}		
if(macroblock_motion_forward macroblock_motion_backward) {		
if(picture_structure == "frame") {		
if(frame_pred_frame_dct == 0)		
frame_motion_type	2	ulmsbf
} else {		
field_motion_type	2	ulmsbf
}		
if((picture_structure == "frame picture") && (frame_pred_frame_dct == 0) && (macroblock_intra/macroblock_pattern)) {		
dct_type	1	ulmsbf
}		
}		

macroblock_modes

図12

【図14】

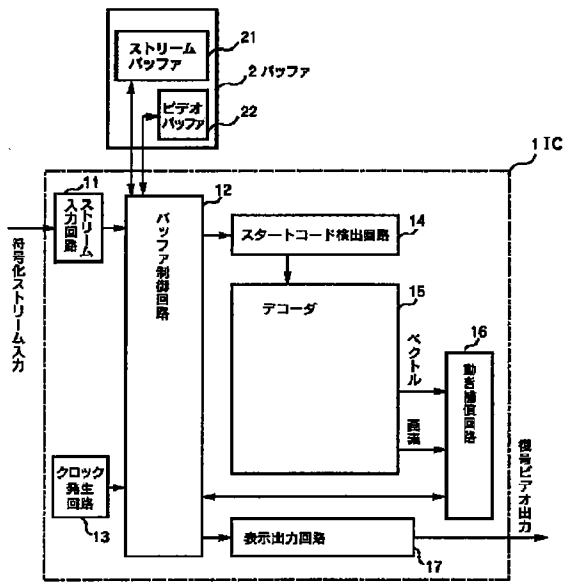


図14

【図16】

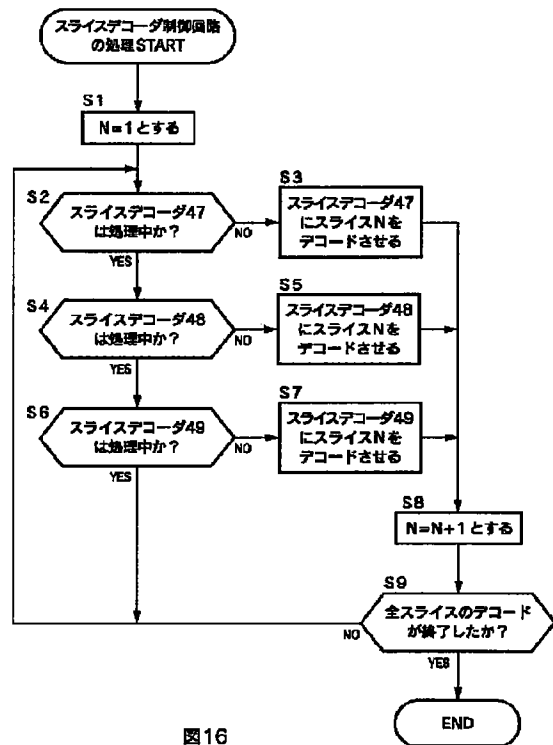


図16

【図17】

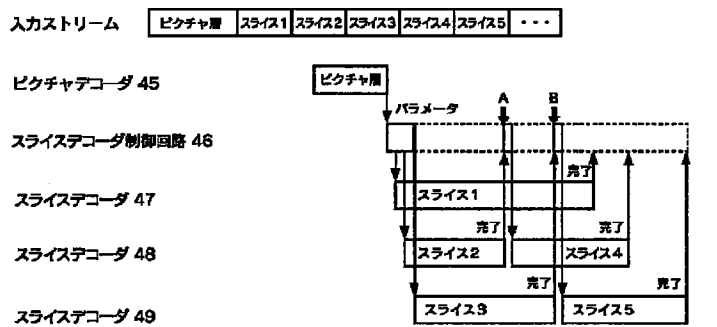


図17

【図15】

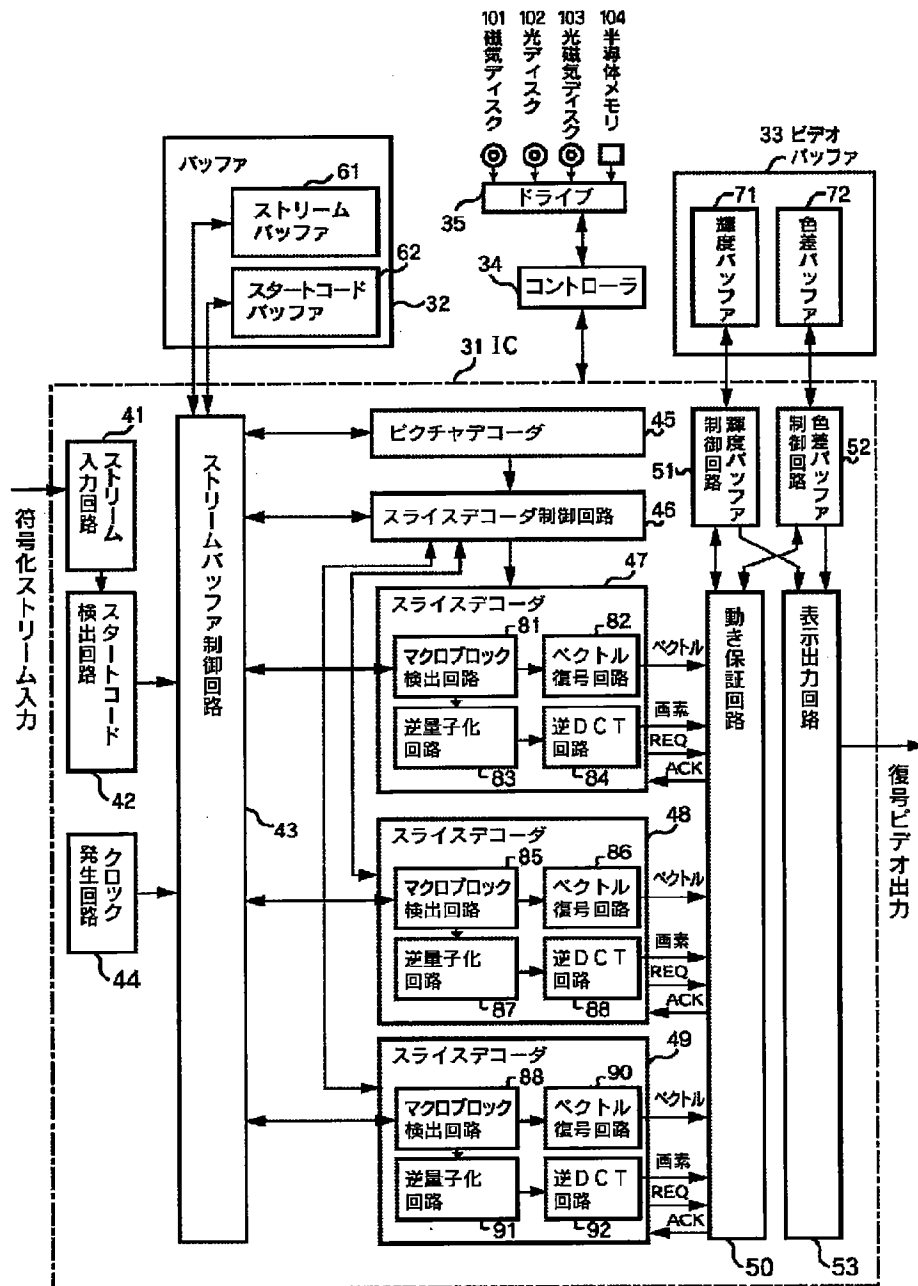


図15

圖18

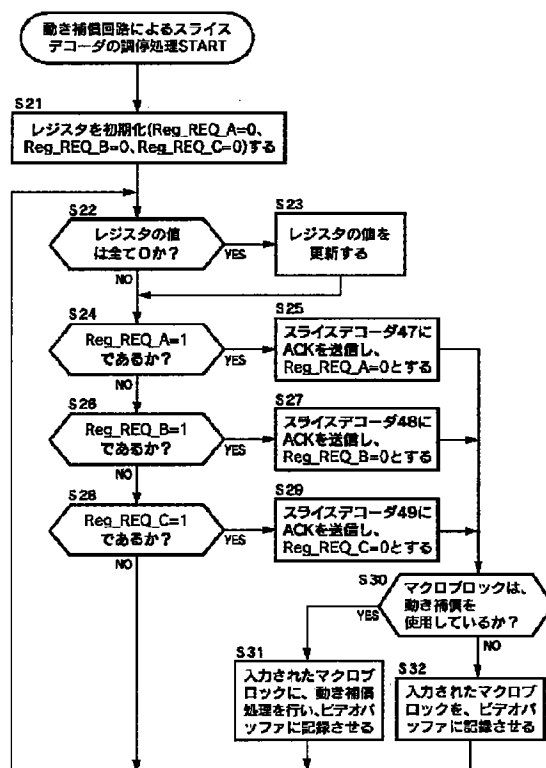
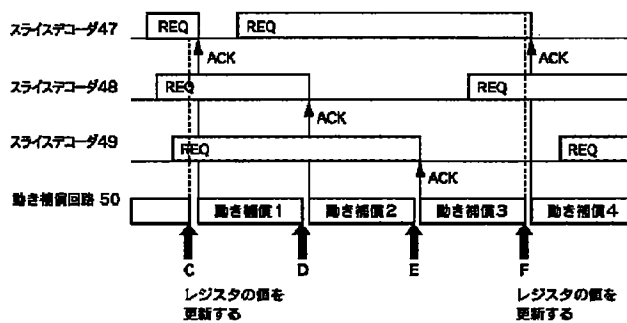


圖 19



【図20】

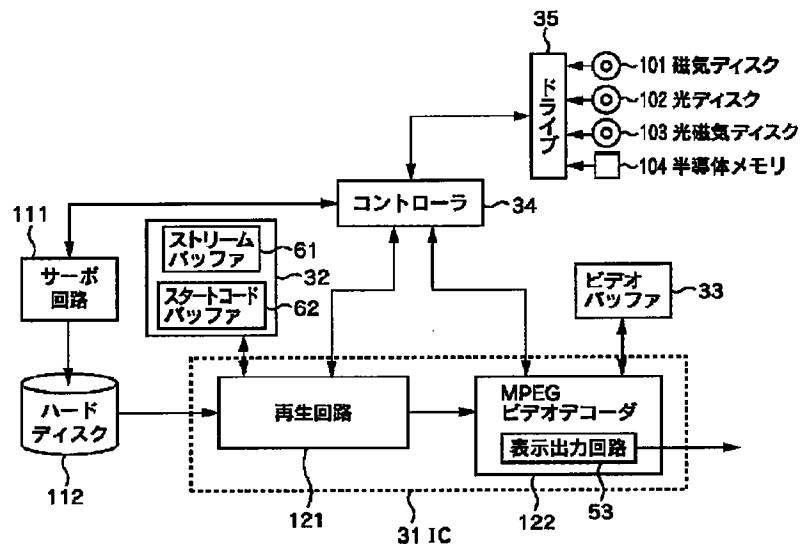
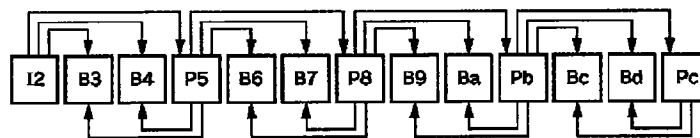


図20

【図21】

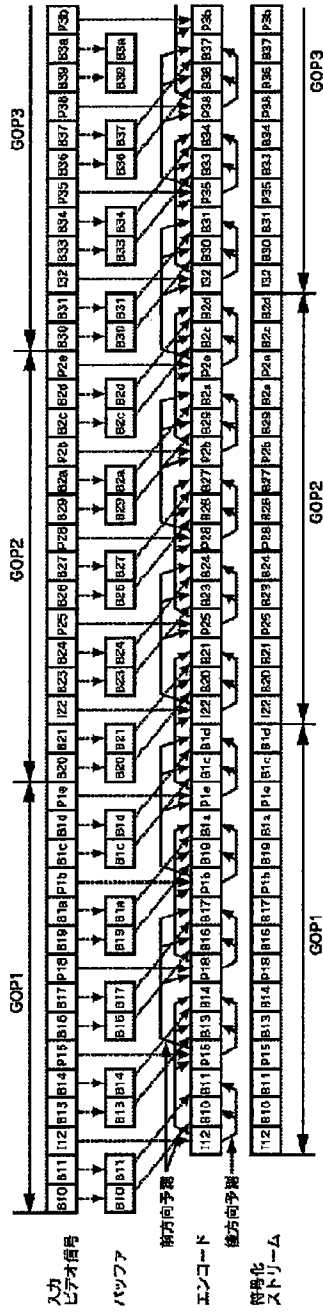
前方向予測



後方向予測

図21

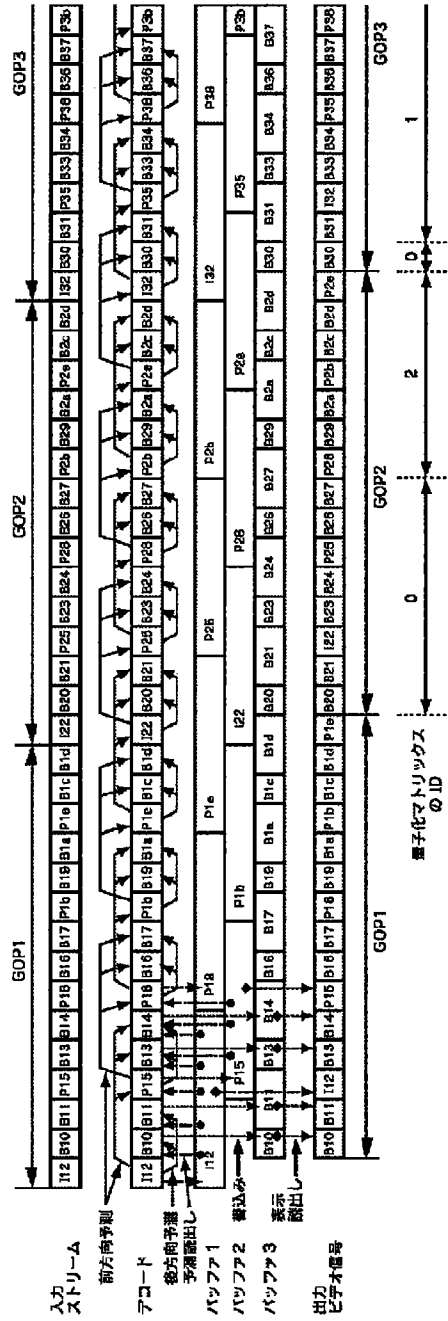
【図22】



MPEG 符号化ストリーム

図22

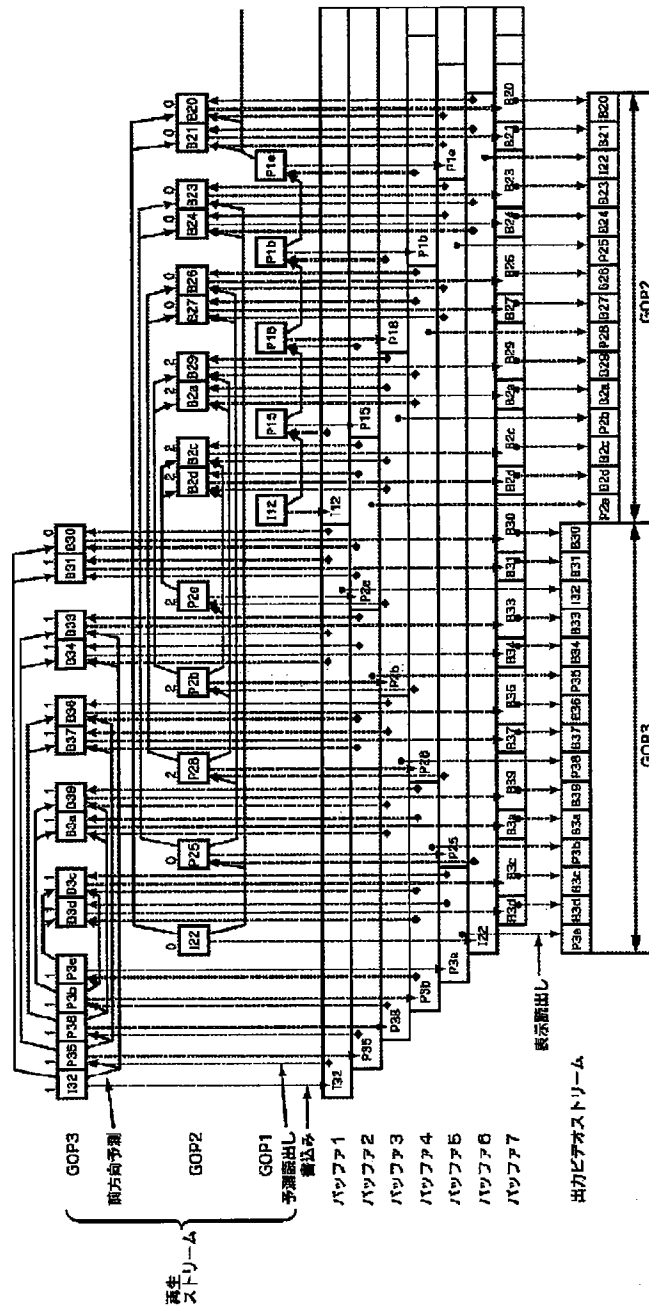
【図23】



フレーム間予測を使用したMPEG 復号例

図23

【図24】



MPEG 逆転再生アコーデの動作

図24

フロントページの続き

(72)発明者 大原 卓巳
 神奈川県川崎市麻生区南黒川10番1号 株
 式会社エルエスアイシステムズ内

(72)発明者 山本 健
 神奈川県川崎市麻生区南黒川10番1号 株
 式会社エルエスアイシステムズ内

(72)発明者 水野 一博
神奈川県川崎市麻生区南黒川10番1号 株
式会社エルエスアイシステムズ内
(72)発明者 森田 年一
神奈川県川崎市麻生区南黒川10番1号 株
式会社エルエスアイシステムズ内

F ターム(参考) 5C053 FA20 GA11 GB38 HA25 KA04
KA05
5C057 AA07 BA01 CA01 DA03 EA02
EC01 EE01 EG05 EM04
5C059 LA09 LB07 MA00 MA05 MA23
MC11 NN01 PP04 PP16 RB02
RC24 SS17 SS18 SS20 UA05
UA31 UA34 UA38
5J064 AA02 AA03 AA04 BA16 BB01
BB03 BC01 BC02 BC16 BD01